

Lappeenrannan teknillinen yliopisto  
Tuotantotalouden osasto  
Diplomityö

## **Tilastollinen tutkimus ohjelmiston laatuun vaikuttavista tekijöistä**

Työn tarkastaja: Professori Tuomo Kässi

Työn ohjaaja: Tekniikan tohtori Ossi Taipale

Espoo, 20.4.2009

Riku Luomansuu  
Lansankallionkuja 4 A 5  
02630 Espoo  
riku.luomansuu@lut.fi  
Puhelin: 040 506 7026

## **TIIVISTELMÄ**

Lappeenrannan teknillinen yliopisto  
Tuotantotalouden osasto

Riku Luomansuu

### **Tilastollinen tutkimus ohjelmiston laatuun vaikuttavista tekijöistä**

Diplomityö

2009

112 sivua, 54 kuvaa, 16 taulukkoa ja 2 liitettä

Tarkastajat: Professori Tuomo Kässi  
Tekniikan tohtori Ossi Taipale

Hakusanat: laatu, ohjelmistotestaus, tilastollinen tutkimus, survey

Tämä työ on tehty osana MASTO-tutkimushanketta, jonka tarkoituksena on kehittää ohjelmistotestauksen adaptiivinen referenssimalli.

Työ toteutettiin tilastollisena tutkimuksena käyttäen survey-menetelmää. Tutkimuksessa haastateltiin 31 organisaatiyksikköä eri puolelta suomea, jotka tekevät keski-kriittisiä sovelluksia.

Tutkimuksen hypoteeseina oli laadun riippuvuus ohjelmistokehitysmenetelmästä, asiakkaan osallistumisesta, standardin toteutumisesta, asiakassuhteesta, liiketoimintasuuntautuneisuudesta, kriittisyydestä, luottamuksesta ja testauksen tasosta.

Hypoteeseista etsittiin korrelaatiota laadun kanssa tekemällä korrelaatio ja regressioanalyysi. Lisäksi tutkimuksessa kartoitettiin minkälaisia ohjelmistokehitykseen liittyviä käytäntöjä, menetelmiä ja työkaluja organisaatiyksiköissä käytettiin, ongelmia ja parannusehdotuksia liittyen ohjelmistotestaukseen, merkittävimpiä tapoja asiakkaan vaikuttamiseksi ohjelmiston laatuun sekä suurimpia hyötyjä ja haittoja ohjelmistokehityksen tai testauksen ulkoistamisessa.

Tutkimuksessa havaittiin, että laatu korreloi positiivisesti ja tilastollisesti merkitsevästi testauksen tason, standardin toteutumisen, asiakasosallistumisen suunnitteluvaiheessa sekä asiakasosallistumisen ohjaukseen kanssa, luottamuksen ja yhden asiakassuhteeseen liittyvän osakysymyksen kanssa. Regressioanalyysin perusteella muodostettiin regressioyhtälö, jossa laadun todettiin positiivisesti riippuvan standardin toteutumisesta, asiakasosallistumisesta suunnitteluvaiheessa sekä luottamuksesta.

## **ABSTRACT**

Lappeenranta University of Technology  
Department of Industrial Management

Riku Luomansuu

### **Statistical Research on the Factors Affecting Software Quality**

Master's Thesis

2009

112 pages, 54 figures, 16 tables and 2 appendices

Supervisors: Professor Tuomo Kässi  
D.Sc. (Tech) Ossi Taipale

Keywords: software testing, software quality, statistical study, survey method

This thesis is a part of the MASTO – research project. The objective of the project was to develop an adaptive reference model for software testing. Survey method was used as the research method. In the study 31 organizational units were interviewed.

The study included hypotheses that software development method, customer participation, compliance of standard, customer relationship, business orientation, criticality of the end-products, trust, and excellence of testing affect quality.

Hypotheses were tested by calculating correlation and regression models. Further, practices, methods and tools used in the OUs were studied.

The results of the correlation study included that quality positively correlates with excellence of testing, compliance of standard, customer participation in development phase, customer participation in general control, and trust. Further, the results of the regression model suggested that quality is associated with compliance of standard, customer participation in planning phase, and trust.

# SISÄLLYSLUETTELO

1	JOHDANTO .....	8
2	TUTKIMUKSEN TARKOITUS JA HYPOTEEESIT .....	10
3	LAATU .....	13
3.1	Laadun määritelmiä .....	13
3.2	Laatuattribuutit .....	14
3.2.1	Toiminnallinen sopivuus .....	15
3.2.2	Luotettavuus .....	16
3.2.3	Suoritustehokkuus .....	16
3.2.4	Käytettävyys .....	16
3.2.5	Turvallisuus .....	17
3.2.6	Yhteensopivuus .....	17
3.2.7	Ylläpidettävyys .....	17
3.2.8	Siirrettävyys .....	18
4	OHJELMISTOTUOTANNOSTA .....	19
4.1	Suunnitelmapohjaiset menetelmät .....	20
4.2	Ketterät menetelmät .....	22
4.3	Yhteenvedo menetelmistä ja niiden eroista .....	24
5	Ohjelmistotestaus .....	27
5.1	Testaus suunnitelmapohjaista ohjelmistokehitysmenetelmää käytettäessä .....	29
5.1.1	Verifioiva testaus .....	30
5.1.2	Validoiva testaus .....	32
5.2	Testaus ketterää ohjelmistokehitysmenetelmää käytettäessä .....	37
6	MUITA OHJELMISTON LAATUUN VAIKUTTAVIA TEKIJÖITÄ .....	41
6.1	Standardin toteutuminen .....	41
6.2	Asiakkaan osallistuminen .....	41
6.3	Asiakassuhteen vaikutus laatuun .....	42
6.4	Luottamus .....	42
6.5	Kriittisyys .....	44
7	TUTKIMUKSEN TOTEUTUS .....	45
7.1	Tilastollinen tutkimus .....	45
7.2	Survey-menetelmä tilastollisen aineiston keräämiseen .....	45
7.2.1	Survey-prosessi .....	46
7.2.2	Reliabiliteetin varmistaminen .....	46
7.2.3	Validiteetin varmistaminen .....	48
7.3	Tulosten analyysiin tarvittavat menetelmät .....	50
7.3.1	Parittainen t-testi .....	51
7.3.2	Cronbachin alfakerroin .....	51
7.3.3	Kolmogorov-Smirnov-testi .....	52
7.3.4	Pearsonin korrelaatiokerroin ja korrelaatiokertoimen neliö .....	52
7.3.5	Lineaarinen regressioanalyysi .....	53
7.3.6	LB (Like Best) tekniikka avointen kysymysten luokitteluun .....	53
8	TUTKIMUKSEN TAUSTATIEDOT .....	54
8.1	Tutkimuksen populaatio ja otos .....	54
8.2	Tutkimuksen rajaus .....	55
8.3	Tutkimusdatan keräämistekniikka .....	55
8.4	Haastattelu-aika .....	57

8.5	Haastatellut henkilöt .....	58
8.6	Haastatellut organisaatioyksiköt.....	59
9	HYPOTEESIEN TESTAUKSEN TULOKSET .....	61
9.1	Liiketoimintasuuntautuneisuus.....	61
9.2	Ohjelmistokehitysmenetelmä .....	62
9.3	Kriittisyys .....	64
9.4	Ohjelmistotestauksen taso .....	66
9.5	Standardin toteutuminen .....	68
9.6	Asiakkaan osallistuminen.....	70
9.7	Asiakassuhde .....	76
9.8	Luottamus.....	78
9.9	Ohjelmiston laatu ja laatuattribuutit .....	80
9.10	Korrelaatiomatriisi ja Kolmogorov-Smirnov .....	82
9.11	Lineaarinen regressiomalli .....	84
10	KARTOITTAVIEN KYSYMYSTEN TULOKSET.....	89
10.1	Systemaattinen menetelmä tai prosessi.....	89
10.2	Käytetyt testaustyökalut .....	90
10.3	Laatujärjestelmäsertifikaatti tai kypsyystasoluokitus .....	91
10.4	Yleisiin laatuattributteihin liittyvät kysymykset.....	92
10.5	Ongelmat ja parannusehdotukset.....	94
10.6	Merkittävimmät tavat asiakkaan ohjelmiston laatuun vaikuttamiseksi .....	96
10.7	Tehokkaimmat testausautomaatioon liittyvää työkalua tai menetelmää ...	98
10.8	Tärkeimmät hyödyt ja haitat ohjelmistokehityksen ja – testauksen ulkoistamisessa.....	101
11	TULOSTEN POHDINTAA.....	104
12	YHTEENVETO .....	107
	LÄHTEET .....	109

LIITE 1. Survey kyselylomake, 7 s.

LIITE 2. Sirona-kuvat, 6 s

## LYHENTEET

ASD	Adaptive Software Development
CMM	Capability Maturity Model
FDD	Feature Driven Development
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization, standardointijärjestö
IEC	International Electrotechnical Commission
LTY	Lappeenrannan teknillinen yliopisto
OU	Organizational Unit, organisaatioyksikkö
PIMS	Profit Impact of Market Strategy
RUP	Rational Unified Process
TKK	Teknillinen korkeakoulu
TPI	Total Process Improvement
XP	eXtreme Programming

## ALKUSANAT

Haluan kiittää Ossi Taipaletta erinomaisesta diplomityöaiheesta sekä erinomaisesta ohjauksesta, joka edesauttoi työn valmistumista jopa hieman suunniteltua aikataulua nopeammin. Kiitokset vaimolleni Sannalle, joka jaksoit kuunnella kaikki lukuisat tilitykset työn viilauksesta ja parantamisesta. Kiitos että jaksoit lukea ja kommentoida tekstiä, olit merkittävässä osassa työn luettavuuden parantamisessa. Kiitokset Muusalle oletettavasti kannustavista örinöistä.

Lopuksi Eerolle haluan sanoa, että ainakin valmistuttiin samalla vuosikymmenellä, vaikka tiukalle se otti.

Espoossa 20.04.2009

Riku Luomansuu

# 1 JOHDANTO

Diplomityö on osa MASTO-hanketta, jossa pyritään kehittämään ohjelmistotestaukselle adaptiivinen referenssimalli, joka voidaan mukauttaa organisaation esimerkiksi liiketoimintasuuntautuneisuuden ja lopputuotteen kriittisyyden perusteella. Tarvella tutkimukselle todettiin Ossi Taipaleen väitöskirjan aikana, (Taipale 2007). Testaukselle on olemassa malleja muun muassa standardien muodossa, mutta ne eivät perustu tutkimukseen vaan yleisesti todettuihin hyviin käytäntöihin. Diplomityön tarkoituksena on selvittää asioita, jotka vaikuttavat ohjelmiston laatuun. Diplomityön tutkimuskysymys on: Mitkä tekijät vaikuttavat ohjelmiston laatuun ja miten?

Diplomityö muodostuu teoriaosuudesta (luvut 1-7) sekä empiirisestä osuudesta (luvut 8-11). Teoriaosuus muodostuu tutkimuksen hypoteesien esittelemisestä ja kappaleista, joissa hypoteesien teoreettinen tausta esitellään tarkemmin. Teoriaosuuteen osuuteen kuuluu myös tilastolliseen analyysiin tarvittavien menetelmien esittely. Empiirisessä osuudessa esitellään työn tulokset.

Diplomityön teoriaosuus on koostettu lähteistä, jotka ovat hyväksi todettuja. Hyvän lähteen tunnistaminen ilman kokemusta on hankalaa, mutta eräs hyvä keino on selvittää, kuinka moni muu henkilö on artikkeleissaan viitannut lähteeseen. Google Scholar (<http://scholar.google.fi/>) on hyödyllinen hakukone tieteellisten artikkeleiden hakemiseen, ja samalla helposti näkee kuinka moni henkilö on haettuun artikkeliin viitannut. Lisäksi oma yli viiden vuoden kokemus ohjelmistoprojektien parissa auttoi hahmottamaan hyviä lähteitä erityisesti ohjelmistotuotantoon ja –testaukseen liittyvissä asioissa. Lisäksi hyvien lähteiden tunnistamisessa olen tukeutunut tohtori Ossi Taipaleen neuvoihin sekä hänen väitöskirjassaan viitattuihin lähteisiin (Taipale, 2007).

Työn lähtökohtana pidetään ISO/IEC:n uutta ohjelmistotestauksen standardia 29119, jonka standardoimistyöhön MASTO-hankkeen projektipäällikkö Ossi Taipale on osallistunut, ja tästä standardista oli eniten saatavilla asiantuntijätietoa.. Eri lähteisiin ja muun muassa 29119 ja 25010 (ohjelmistotuotteen laatu) standardeihin pohjautuen



Lappeenrannan teknillisen yliopiston MASTO-hankkeeseen osallistuneet henkilöt olivat rakentaneet tarvittavan survey-lomakkeen tutkimushypoteesien testaamista varten. Näin ollen diplomityön teoriaosuus rajoittuu survey-lomakkeessa käsiteltäviin asioihin.

On myös hyvä huomata, että testauksella ei ole mitään yleisesti vakiintunutta teoriaa, joka olisi voinut toimia työn lähtökohtana (Kitchenham et al., 2002). Kansainvälistä standardia voidaan pitää hyvänä lähtökohtana, koska standardi on kokoelma kansainvälisesti ohjelmistoteollisuudessa hyväksytyistä käytännöistä (Reid, 2008). Standardia ei kuitenkaan voi nimittää teoriaksi. Tätä lähestymistapaa puoltaa ote Hirsjärven kirjasta: ”Jos teoriaa ei löydy, olisi vältettävä keinotekoisia ratkaisuja, kuten esimerkiksi minkä tahansa ajatusrakennelman nimittämistä teoriaksi” (Hirsjärvi et al., 2004).”

## 2 TUTKIMUKSEN TARKOITUS JA HYPOTEESEIT

Tutkimuksen tarkoitus voidaan jakaa Hirsjärven mukaan kartoittavaan, selittävään, kuvailevaan tai ennustavaan tutkimukseen (Hirsjärvi et al., 2004). Näistä vaihtoehdoista tämä tutkimus sijoittuu selittävän ja kartoittavan tutkimuksen välimaastoon. Taulukossa 1 on kuvattu tämän tyyppiseen tutkimukseen liittyvä tutkimuskysymys ja -strategia. Tutkimusstrategiaksi valittiin kvantitatiivinen eli tilastollinen tutkimus, joka toteutettiin survey-kyselytutkimuksen avulla.

”Yleinen käytäntö on, että selittävässä ja vertailevassa tutkimuksessa käytetään hypoteeseja. Hypoteesilla tarkoitetaan ennakoitavaa ratkaisua tai selitystä”, (Hirsjärvi et al., 2004) Tämän tutkimuksen hypoteesit olivat peräisin Ossi Taipaleen (Taipale, 2007) väitöskirjasta, missä ne ovat eritelty nimellä tutkimuskohteet (research issues) sekä MASTO-hankkeessa väitöskirjatutkimusta tehneen Aparna Pasin hankkimista lähteistä.

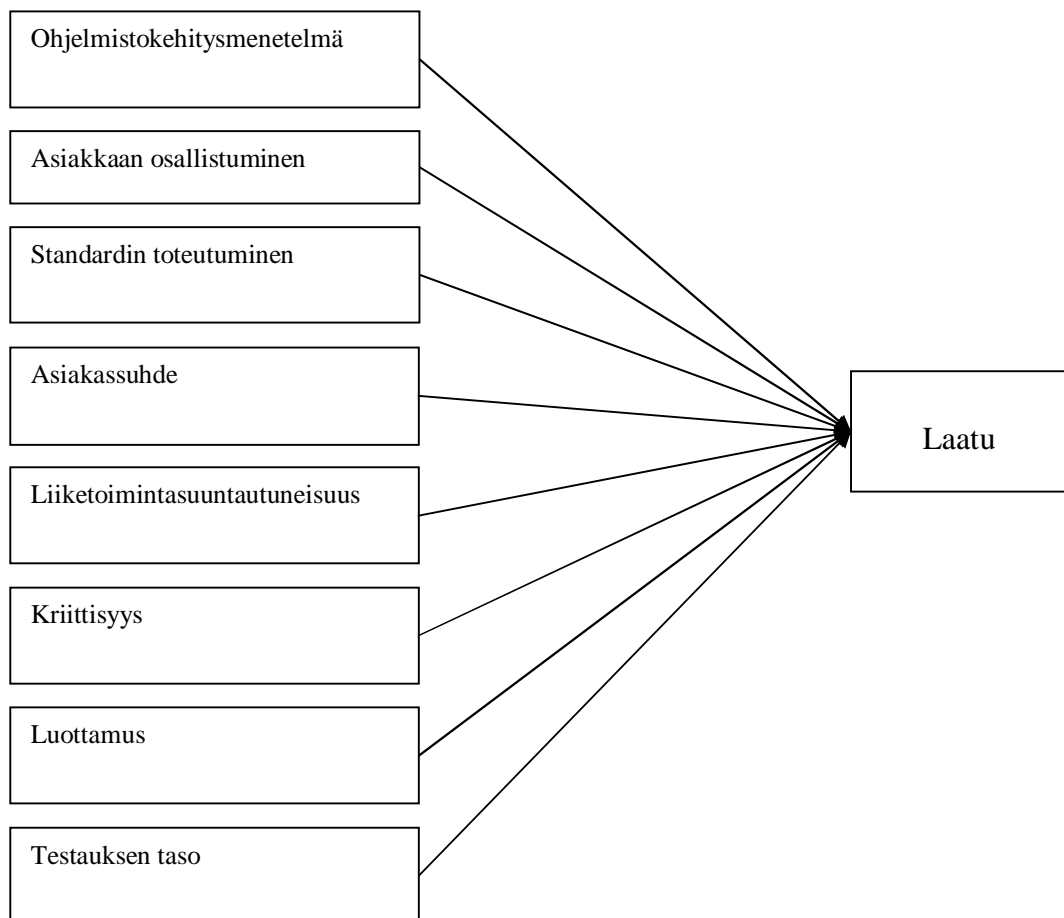
Hypoteesit ristiintarkistettiin Standardia 29119 vasten, jolloin saatiin varmistettua, että hypoteesit ovat sidoksissa johonkin ulkopuoliseen viitekehykseen. Toisin sanoen

**Taulukko 1. Tutkimuksen tarkoituksen liittyminen tutkimuskysymykseen ja tutkimusstrategiaan, muokattu lähteestä (Hirsjärvi et al., 2004)**

Tutkimuksen tarkoitus	Tutkimuskysymys	Strategia
<ul style="list-style-type: none"> <li>Etsiä selitystä tilanteelle tai ongelmaan</li> <li>Tunnistaa todennäköisiä syy-seurausketjuja</li> <li>Katsoa mitä tapahtuu</li> <li>Etsiä uusia näkökulmia, löytää uusia ilmiöitä</li> <li>Kehittää hypoteeseja</li> </ul>	<ul style="list-style-type: none"> <li>Mitkä tapahtumat, uskomukset, asenteet ja toiminnat ovat vaikuttaneet tähän ilmiöön? Kuinka nämä tekijät ovat vuorovaikutuksessa keskenään?</li> <li>Mitkä ovat keskeiset teemat, mallit luokat? Miten tyypittelyt ovat suhteessa toisiinsa?</li> </ul>	<ul style="list-style-type: none"> <li>Voi olla kvantitatiivinen tai kvalitatiivinen</li> <li>Kenttätutkimus</li> <li>Tapaustutkimus</li> <li>Historialliset menetelmät</li> </ul>

hypoteesien voidaan katsoa olevan perusteltuja. Hirsjärven (Hirsjärvi et al., 2004) mukaan mikäli perusteluja hypoteeseille ei löydy, ”olisi luovuttava hypoteesien asettamisesta”.

Kuvassa 1 on esitetty tutkimuksen hypoteesit. Tutkimusryhmän tavoitteena oli tutkia, pitävätkö hypoteesit paikkansa.



**Kuva 1. Tutkimuksen hypoteesit (laatuun vaikuttavia tekijöitä)**

Seuraavissa kappaleissa selitetään tutkimuksen teoreettinen tausta määrittelemällä laadun ja hypoteesien teoreettinen tausta. Survey-lomakkeen kysymysten lähde hypoteesien mittaamiseksi on mainittu kussakin hypoteesin teoreettista taustaa käsittelevässä kappaleessa. Mikäli lähde ei ole mainittu, ovat kysymykset tutkimusryhmän itse kehittämiä.

Teoreettisessa osassa on painoarvoa annettu ohjelmistokehitysmenetelmä- sekä testauksen taso-hypoteeseille teoreettisen taustan selvittämiseksi, koska niillä on laatuun välitön vaikutus. Muut hypoteesit vaikuttavat laatuun välillisesti ohjelmistokehitysmenetelmän ja testauksen kautta.

### 3 LAATU

Seuraavissa kappaleissa esitetään laadun määritelmiä sekä tutkimuksessa käytetyn standardin ISO/IEC 25010 määrittelemät laatuattribuutit.

#### 3.1 Laadun määritelmiä

Termillä ”laatu” on olemassa useita eri määritelmiä. Esimerkiksi Haikala & Märijärvi teoksessaan Ohjelmistotuotanto (Haikala ja Märijärvi, 2006) ovat keränneet eri lähteistä 10 eri laadun määritelmää, taulukko 2.

Yksi Haikalan ja Märijärven mainitsema tapa laadukkuuden määrittelemiseksi on jakaa laatu kolmeen eri komponenttiin: objektiivisesti arvioitavaan komponenttiin,

**Taulukko 2. Laadun määritelmiä, (Haikala ja Märijärvi, 2006)**

Määrittelijä	Määritelmä
Suomen kielen perus-sanakirja	Se mikä on jollekin ominaista, jonkin ominaisuudet, luonne, olemus, kvaliteetti.
Webster's New 20th Century Dictionary	Superiority, excellence, that which belongs to something and makes or helps to make it what it is.
Oxford dictionary	Erinomaisuuden aste.
ISO	Se, missä määrin luontaiset ominaisuudet täyttävät vaatimukset.
Deming	Asiakkaalle tärkein tuote.
Crosby	Täyttää vaatimukset.
Ishikawa	Asiakastyytyväisyys.
Wesselius	Laatu = objektiivisesti arvioitavissa oleva komponentti + subjektiivisesti arvioitavissa oleva komponentti + kokonaan arvioimattomissa oleva komponentti.
PIMS-tietokanta	Laatu: hintaa lukuun ottamatta kaikki muut ostopäätökseen vaikuttavat attribuutit.
Kauppamiehen aksiooma	Asiakastyytyväisyys on tärkeintä. Tyytyväiset asiakkaat ostavat uudestaan ja kertovat muillekin hyvästä tuotteesta. Hyvä tuote tarkoittaa parempaa kuin kilpailijoilla.

subjektiivisesti arvioitavaan komponenttiin ja kokonaan arvioimattomissa olevaan komponenttiin (Wesselius ja Ververs, 1990).

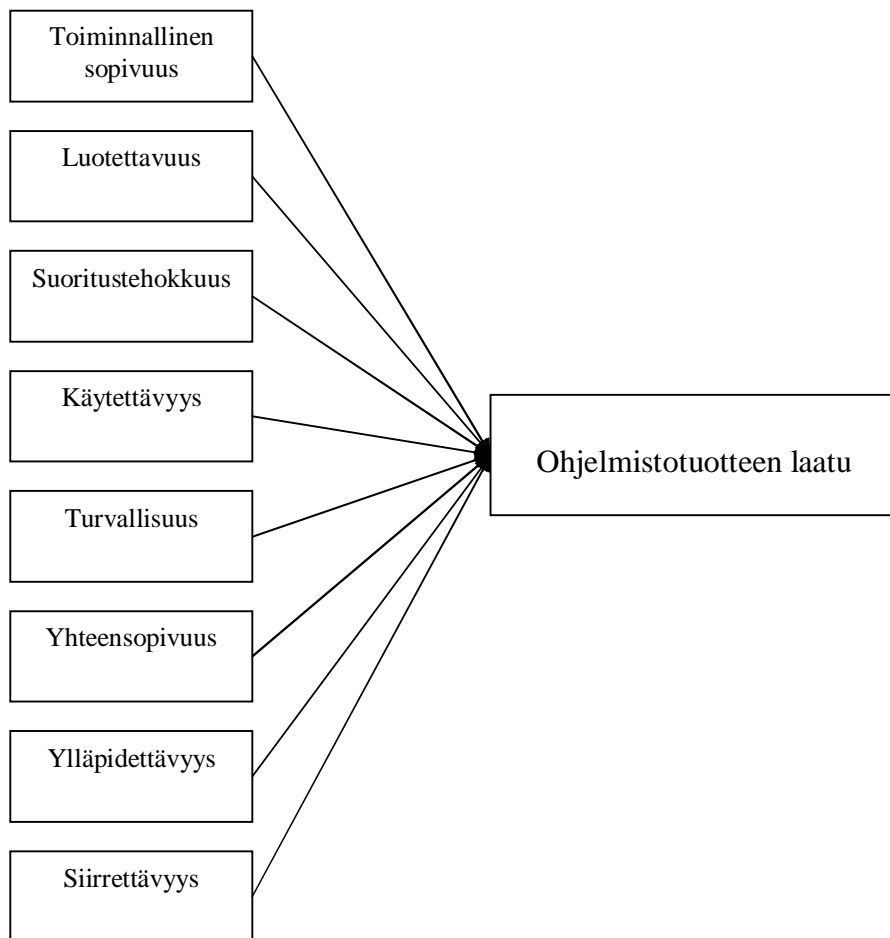
Objektiivisella laatukomponentilla tarkoitetaan, kuinka toteutettu tuote toteuttaa kuvatut vaatimukset. Subjektiivinen laatukomponentti taas tarkoittaa järjestelmän käyttäjän subjektiivisesti arvioimaa laatua, eli. kuinka hyvin järjestelmä täyttää asiakkaan odotukset. Objektiivinen laatukomponentti voi siis olla korkealla tasolla, vaikka subjektiivisen laatukomponentin perusteella järjestelmä ei tee sitä mitä käyttäjä haluaisi sen tekevän. Tässä tapauksessa ongelmana on huono vaatimusten dokumentointi.

Arvioimattomissa olevalla komponentilla tarkoitetaan toimitushetkellä piilossa olevaa laatua, eli kuinka järjestelmää pystytään muuttamaan asiakkaiden vielä piilossa olevien tulevien tarpeiden mukaiseksi. Tällä komponentilla on merkitystä, sillä esimerkiksi asiakkaan kanssa mahdollisia jatkokauppoja on helpompi solmia, mikäli tuotetta pystyy helposti muokkaamaan asiakkaan tarpeiden mukaiseksi. Muunneltavuus pystytään huomioimaan esimerkiksi järjestelmän arkkitehtuuria suunniteltaessa.

Tässä tutkimuksessa laatu määritelmänä on käytetty standardin ISO/IEC 25010 mukaista (ISO/IEC, 2008) laadun määritelmä, joka on vapaasti käännettynä: ”Ohjelmiston laatu on määrä, jolla ohjelmistotuote tyydyttää suorat ja epäsuorat sille asetetut tarpeet määritetyssä ympäristössä”.

### **3.2 Laatuattribuutit**

Standardissa ISO/IEC 25010 (ISO/IEC, 2008) laatu on jaettu kahdeksaan laatuattribuuttiin kuvan 2 mukaisesti. Lisäksi jokainen laatuattribuutti on standardissa jaettu alilaatuattribuutteihin, jotka kuvaat asioita, joista laatuattribuutti voi koostua. Jokaisen laatuattribuutin aliattribuuttina standardissa on noudatettavuus (compliance), joka tarkoittaa kuinka hyvin laatuattribuutti noudattaa olemassa olevia standardeja sekä hyviä käytäntöjä.



**Kuva 2. Laadun koostuminen laatuattribuuteista, ISO/IEC 25010 (ISO/IEC, 2008)**

Seuraavissa kappaleissa selvitetään mitä laatuattribuutit tarkoittavat. Attribuuttien aliattribuuttien merkitystä ei ole selitetty yksityiskohtaisesti, koska standardissa mainitaan, että standardin soveltaja voi lisätä omia alilaatuattribuutteja, mikäli se katsotaan tarpeelliseksi. Tämä viittaa siihen, että laatuattribuutin tulkinnassa on liikkumavaraa, eikä standardissa lueteltu alilaatuattribuuttalista ole kattava. Näin ollen ainoastaan laatuattribuuttien yleinen merkitsevyys on relevanttia tietoa tutkimuksen kannalta.

### 3.2.1 Toiminnallinen sopivuus

Toiminnallisella sopivuudella tarkoitetaan, kuinka hyvin ohjelma sopii siihen ympäristöön mihin se on tehty. ISO/IEC 25010 (ISO/IEC, 2008) määrittelee sen tarkoittavan ” The degree to which the software product provides functions that meet stated

and implied needs when the software is used under specified conditions.” Wesseliuksen määritelmän mukaan kyseessä subjektiivinen laatukomponentti, koska sitä ei pysty objektiivisesti arvioimaan (Wesselius ja Ververs, 1990).

### **3.2.2 Luotettavuus**

Luotettavuudella tarkoitetaan tässä sitä, kuinka luottavasti ohjelma toimii. Standardi määrittelee luotettavuuden koostuvan seuraavista asioista: saatavuus (availability), vikasietoisuus ja palautuvuus. Saatavuus tarkoittaa, kuinka suuren osan ajasta järjestelmä on käytettävissä. Aika määritellään yleensä vuositasolla prosenttilukuna. Ideaalitulanteessa lukema on 100 %. Tähän on kuitenkin käytännössä lähes mahdotonta päästä. Vikasietoisuudella taas tarkoitetaan sitä, kuinka hyvin järjestelmä pystyy toimimaan vian sattuessa järjestelmän jossakin osassa. Palautuvuudella tarkoitetaan sitä, kuinka hyvin järjestelmä pystyy palautumaan tilaan ennen vian sattumista. Wesseliuksen (Wesselius ja Ververs, 1990) määritelmän mukaan kyseessä on objektiivinen laatukomponentti, koska sen pystyy tarvittaessa mittaamaan.

### **3.2.3 Suoritustehokkuus**

Suoritustehokkuus on standardin mukaan sitä, kuinka tehokkaasti ohjelma hyödyntää käyttöympäristönsä resurssit (muistin, prosessoritehon, jne.) ja kuinka tehokkaasti se toimii käyttäjän näkökulmasta esimerkiksi vasteaikojen muodossa. Wesseliuksen (Wesselius ja Ververs, 1990) määritelmän mukaan myös tämä on objektiivinen laatukomponentti, koska sen pystyy tarvittaessa mittaamaan.

### **3.2.4 Käytettävyys**

Käytettävyys kuvaa standardin mukaan esimerkiksi ohjelman helppokäyttöisyyttä, kuinka hyvin ohjelma neuvoo käyttäjiä apua tarvittaessa ja kuinka puoleensavetävä ohjelman käyttöliittymä on. Wesseliuksen (Wesselius ja Ververs, 1990) määritelmän



mukaan tämä on subjektiivinen laatukomponentti, koska käytettävyyden ”hyvyys” on arvioijan mielipiteestä kiinni.

### **3.2.5 Turvallisuus**

Turvallisuudella tarkoitetaan turvanäkökohtien huomioonottamista esimerkiksi siten, että järjestelmään eivät ulkopuoliset pysty tunkeutumaan, käsiteltäviä tietoja ei näe kuka tahansa ja järjestelmän tietoja ei pystytä poistamaan vahingossa/tarkoituksellisesti. Wesseliuksen (Wesselius ja Ververs, 1990) määritelmän mukaan tämä on objektiivinen laatukomponentti, koska tämä pystytään arvioimaan riippumatta arvioijasta.

### **3.2.6 Yhteensopivuus**

Yhteensopivuus merkitsee yhteensopivuutta muiden ohjelmistojen ja ohjelmistokomponenttien kanssa. Sillä tarkoitetaan esimerkiksi sitä, pystyykö ohjelmisto toimimaan samassa asennusympäristössä muiden ohjelmien kanssa. Wesseliuksen (Wesselius ja Ververs, 1990) määritelmän mukaan myös tämä on objektiivinen laatukomponentti, koska tämä pystytään arvioimaan riippumatta arvioijasta.

### **3.2.7 Ylläpidettävyys**

Ylläpidettävyydellä tarkoitetaan, kuinka helposti ohjelmaa voidaan muokata asennuksen jälkeen. Siihen vaikuttavat muun muassa kuinka helppoa on tehdä muutoksia ilman että jotain muuta osaa rikotaan, muokkausten testattavuuden helppous, ohjelmiston vikojen analysoinnin helppous jne. Wesseliuksen (Wesselius ja Ververs, 1990) määritelmän mukaan tämä on arvioimattomissa oleva laatukomponentti, koska arviointia ei pystytä tekemään täsmälleen ostohetkellä. Tässä attribuutissa on myös subjektiivisen laatukomponentin piirteitä, koska tätä pystytään osittain arvioimaan ostohetkellä. Sen sijaan objektiivinen mittaus tälle attribuutille on vaikeaa.

### 3.2.8 Siirrettävyys

Siirrettävyydellä tarkoitetaan sitä, kuinka helppoa ohjelmiston siirtäminen on laitteisto- tai ohjelmistoympäristöstä toiseen ympäristöön. Tämä attribuutti sisältää myös ohjelmiston skaalautuvuuden suuremmaksi tai pienemmäksi esimerkiksi käyttäjämäärien suhteen. Wesseliuksen (Wesselius ja Ververs, 1990) määritelmän mukaan tämä on subjektiivinen laatukomponentti, koska tätä ei pystytä yksikäsitteisesti mittaamaan.

Huolimatta siitä että moni attribuutti olisi objektiivisesti arvioitavissa, on tutkimuksessa kaikki laatuattribuutit kuitenkin arvioitu subjektiivisesti, koska tutkimus toteutettiin haastattelemalla. Objektiivinen arviointi olisi ollut hankalaa, koska kyseisiä attribuutteja ei yrityksen ulkopuolinen henkilö pääse helposti mittaamaan.

## 4 OHJELMISTOTUOTANNOSTA

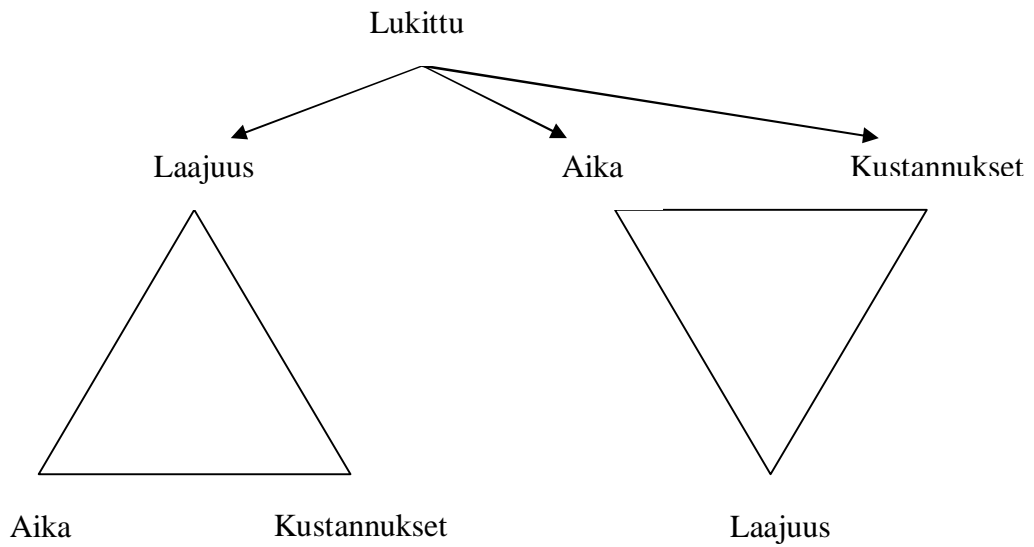
Sommervillen (Sommerville, 1995) mukaan ohjelmistotuotteet voidaan jakaa kahteen laajaan luokkaan: yleistuotteisiin ja räätälöityihin tuotteisiin. Ossi Taipaleen väitöskirjassa (Taipale, 2007) on jaotteluun lisätty vielä alihankinta sekä konsultointi, jolloin yrityksen liiketoimintasuuntautuneisuudeksi saadaan kuvan 3 mukainen jatkumo.



**Kuva 3. Liiketoimintasuuntautuneisuus, (Taipale, 2007)**

Liiketoimintasuuntautuneisuudella on vaikutusta siihen miten testausta tehdään, (Taipale, 2007). Näin ollen sillä on mahdollisesti vaikutusta laatuun ja se halutaan siksi tässä tutkimuksessa selvittää. Liiketoimintasuuntautumisen yksityiskohtaisempi kuvaaminen on rajattu tämän diplomityön ulkopuolelle.

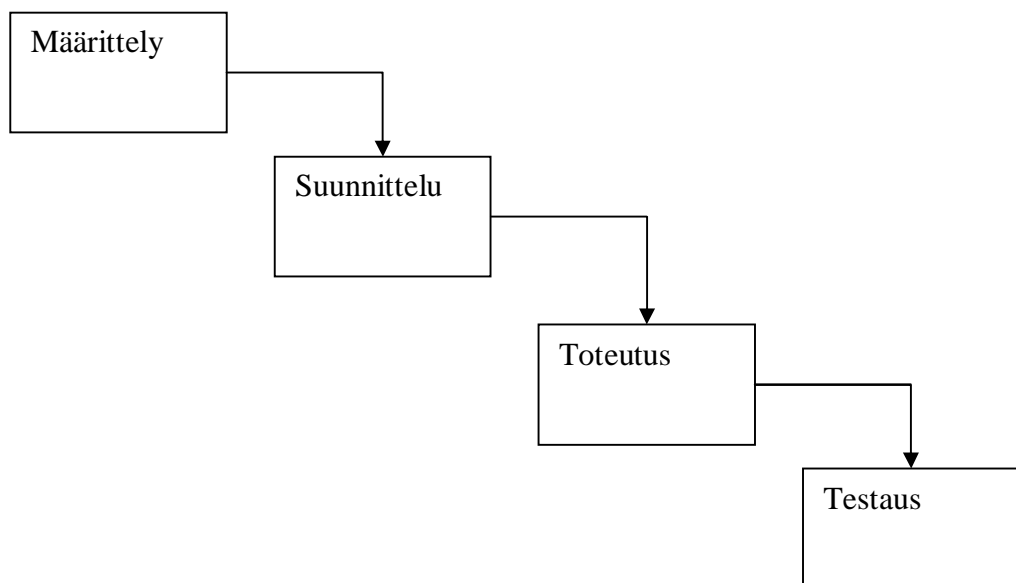
Ohjelmistotuotantomenetelmät ohjelmistotuotteiden tuottamiseksi voidaan jakaa suunnitelmapohjaisiin ja ketteriin menetelmiin (Boehm ja Turner, 2004). Liiketoiminnalliselta kannalta katsottuna ohjelmistojen kehittämisessä on kolme muuttujaa: laajuus, aika ja kustannus. Suunnitelmapohjaista menetelmää käytettäessä laajuus on lukittu, kun taas kehitykseen käytettävä aika ja kustannukset ovat muuttuvia. Ketteriä menetelmiä käytettäessä laajuus on muuttuva kun taas kehitykseen käytettävä aika ja kustannus ovat lukittuja. Kuva 4 kuvaa tilannetta.



Kuva 4. Muuttujat liiketoiminnallisesta näkökulmasta ohjelmistoa kehittäessä

#### 4.1 Suunnitelmapohjaiset menetelmät

Suunnitelmapohjaisissa menetelmissä pyritään suunnittelemaan etukäteen toteutettava ohjelmisto mahdollisimman pitkälle. Suunnittelun edellytyksenä on, että vaatimukset ovat suhteellisen stabiileja. Suunnitelmapohjaiset menetelmät noudattavat yleisimmin vesiputousmallia, joka yksinkertaisimmillaan on kuvattu kuvassa 5.



Kuva 5. Vesiputousmalli

Ideana vesiputousmallin mukaisessa kehityksessä on ensin määritellä tarkasti mitä toteutetaan. Määrittely on syötteenä Suunnittelu-vaiheelle, jossa suunnitellaan määrittelyyn perustuen miten järjestelmä toteutetaan. Suunnittelu-vaihe taas toimii syötteenä Toteutus-vaiheelle, jossa Suunnittelun perusteella järjestelmä ohjelmoidaan. Toteutus toimii taas syötteenä testausvaiheelle, jossa järjestelmää testataan.

Vesiputousmalleja on kehitetty useita eri versioita, jotka laajentavat alkuperäistä mallia. Esimerkiksi teoksessa Ohjelmistotuotanto (Haikala ja Märijärvi, 2006) on testaus-vaiheen perään lisätty erillisenä vaiheena vielä käyttöönotto ja ylläpito. Oleellista suunnitelmapohjaisen menetelmän käytössä on jokaisen vaiheen dokumentointi ja vaiheesta toiseen liikuttaessa dokumenttien toisistaan johdettavuus toisin sanoen jäljitettävyyden. Suunnitelmapohjaista menetelmää käytettäessä pitäisivät vaatimukset olla mahdollista jäljittää aina kooditasolle asti. Haittapuolena on menetelmän jäykkyys erityisesti silloin, kun ohjelmiston vaatimukset muuttavat paljon matkan varrella, koska muutokset täytyy valuttaa ”vesiputousta” pitkin jokaisen vaiheen dokumentteihin.

Vesiputousmallien lisäksi on olemassa myös muita suunnitelmapohjaisia menetelmiä, esimerkiksi evo-malli ja erilaiset iteratiiviset mallit. Yhteistä vesiputousmallien kanssa näissä on suuri dokumentaatio ja jäljitettävyyden vaatimukset, (Boehm ja Turner, 2004). Suunnitelmapohjaisiin menetelmiin liittyy myös tiukka prosessien noudattaminen ja pyrkimys prosessien hallintaan esimerkiksi mittareiden avulla. Näin ollen esimerkiksi henkilöiden roolit ovat tarkasti määriteltyjä. Prosessien kehittämiseen on olemassa erilaisia kehittämismalleja, esimerkiksi CMMi, (Software Engineering Institute, 2009). Suunnitelmapohjainen menetelmä edellyttää myös tiukkoja sopimussellisia sitoumuksia siitä, mitä asiakkaalle tuotetaan (Boehm ja Turner, 2004).

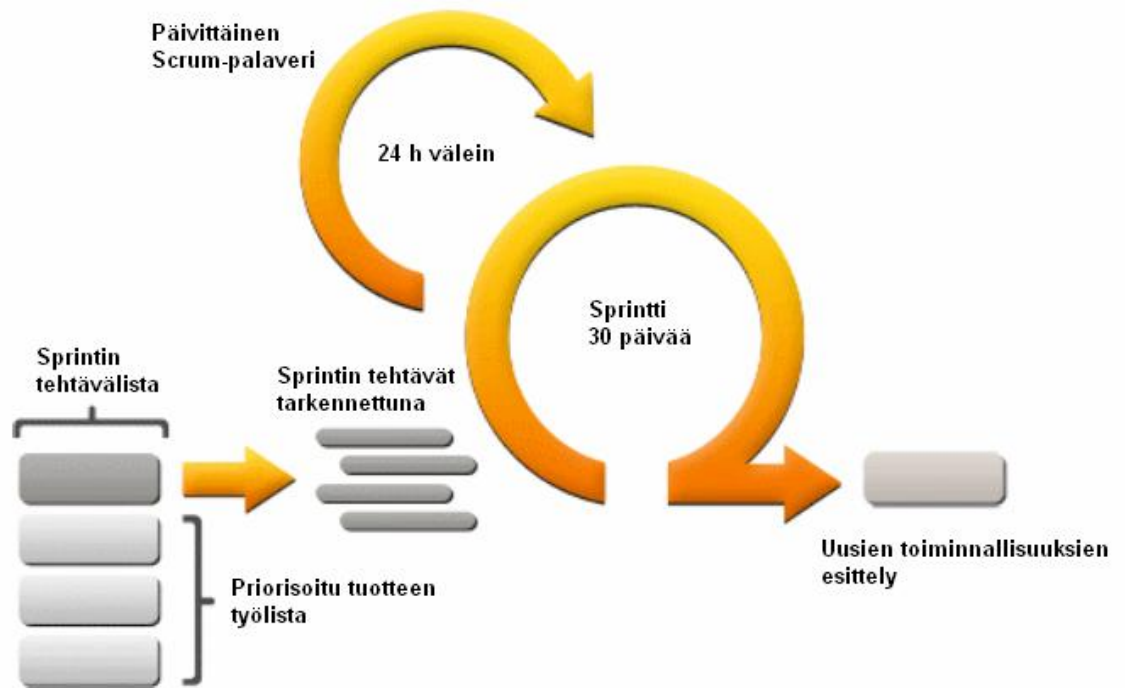
Suunnitelmapohjaisia menetelmiä käytetään suurissa projekteissa, joissa ohjelmistokehittäjien määrä on suuri. Tämä johtuu siitä, että tiedon välitys suurissa mittakaavoissa ei toimi muulla tavoin kuin dokumentoimalla. Myös järjestelmän arkkitehtuuri on suurissa projekteissa suunniteltava etukäteen, jotta järjestelmä saataisiin toimivaksi.

## **4.2 Ketterät menetelmät**

Ketterien menetelmien pyrkimyksenä on päästä eroon suunnitelmapohjaisten menetelmien raskaudesta, ja muutosten tekeminen näitä menetelmiä käytettäessä pitäisi olla helpompaa. Ketterissä menetelmissä dokumentaatio pyritään pitämään mahdollisimman pienenä, ja esimerkiksi tiedon välittäminen ohjelmistokehittäjien kesken hoidetaan suoraan kasvokkain keskustelemalla. Esimerkiksi Scrum-menetelmässä, josta minulla on käytännön kokemusta, tiedon välitys kehittäjien kesken hoidetaan päivittäisissä palavereissa, joissa käydään läpi kolme kysymystä: 1) Mitä olet tehnyt viime tapaamisen jälkeen, 2) Mitä teet ennen seuraavaa tapaamista ja 3) Onko mitään esteitä tai hankaluuksia, jotka estävät työn tekemisen.

Asiakkaan vaatimukset on kuvattu niin sanottuina käyttäjätarina (user stories), jotka ovat tapa kuvata loppukäyttäjän kannalta haluttuja toimintoja ilman suurta dokumenttimäärää. Käyttäjätarinat pyritään pitämään kompakteina, mistä johtuen ne ovat toteuttajan kannalta tulkinnanvaraisia. Jotta toteutuksesta tulisi haluttu, on asiakkaan oltava tiiviissä yhteistyössä toteuttajien kanssa. Näin toteuttaja pystyy tarvittaessa kysymään selvennystä käyttäjätarinan tulkintaan.

Ketterissä menetelmissä ei teoriassa suunnitella mitään etukäteen vaan kaikki toteutettavat asiat tehdään niin sanotussa iteraatioissa, jotka ovat tyypillisesti kahden viikon mittaisia. Iteraation alussa iteraation aikana toteutettavat asiat päätetään yhdessä asiakkaan kanssa. Asiakkaalle iteraation lyhyestä pituudesta on hyötyä, koska tällöin asiakkaalla mahdollisuus priorisoida sellaisia asioita, mitkä hänen/heidän näkökulmastaan iteraation alkamishetkellä tuottavat eniten lisäarvoa. Kuvassa 6 on esitettyä Scrum-menetelmän prosessi, jossa menetelmän terminologian mukainen sana ”sprintti” tarkoittaa yhtä iteraatiota.



Kuva 6. Scrum-menetelmän prosessi, (Koskela, 2007)

Jatkuva priorisointi edellyttää myös paljon tiiviistä yhteistyötä asiakkaan kanssa, koska asiakkaat viime kädessä määräävät prioriteetin. Luottamus asiakkaan ja toimitajan välillä on tärkeää, koska menetelmän lyhyiden iteraatioiden vuoksi kaikkia asioita ei voida sopia tiukasti sopimusteitse etukäteen, (Boehm ja Turner, 2004). Myös hyväksymistestausta tehdään jokaisen iteraation päätteeksi, joten tässäkin mielessä tiivis yhteistyö on välttämätöntä.

Yksi ketterien menetelmien perusfilosofioista on tehdä arkkitehtuuri niin yksinkertaiseksi kuin mahdollista – Boehm puhuu kirjassaan YAGNI (You Are not Going to Need It) ohjenuorasta (Boehm ja Turner, 2004). Ohjenuoran mukaan arkkitehtuuri on tehtävä niin yksinkertaiseksi kuin mahdollista, koska suunnittelu tulevaisuutta varten vaatimusten jatkuvasti muuttuessa on resurssien tuhlaamista. Tästä seurauksena on se, että ketterät menetelmät eivät sovi kovin hyvin suurien järjestelmien kehittämiseen, koska arkkitehtuurin etukäteen suunnittelemattomuus saattaa aiheuttaa, että haluttuja lisäyksiä ei voida myöhemmin kohtuullisesti toteuttaa valitun arkkitehtuurin takia.

Toinen ongelma ketterien menetelmien soveltamisessa suuriin järjestelmiin on tiedon jakaminen ohjelmiston kehittäjien kesken. Pienessä mittakaavassa tiedon jakaminen päivittäisissä palavereissa onnistuu, mutta henkilömäärän kasvaessa tiedon jakaminen ilman dokumentaatiota vaikeutuu.

Ketteriä menetelmiä käytettäessä osaamisvaatimukset keskimääräiselle ohjelmistokehittäjälle ovat kovemmat kuin suunnitelmapohjaisia menetelmiä käytettäessä. Tämä johtuu siitä, että suunnitelmapohjaisia menetelmiä käytettäessä pystytään järjestelmän arkkitehtuuri suunnittelemaan ennakkoon ja jakamaan toteutettavat osat osaamisen mukaan. Järjestelmän toiminta on tällöin myös dokumentoitu. Ketterissä menetelmissä yksittäisen ohjelmistokehittäjän pitää tarkemmin ymmärtää, mitä hän on tekemässä ja miten koodin muuttaminen vaikuttaa muihin osiin, koska tarkkoja määrittelyjä ei ole.

Ketteriä menetelmiä on useita, joista edellä mainittu Scrum on tunnetuimpia. Muita ketteriä menetelmiä ovat esimerkiksi XP (Extreme Programming), ASD (Adaptive Software Development), Crystal ja FDD (Feature Driven Development), (Boehm ja Turner, 2004).

### ***4.3 Yhteenveto menetelmistä ja niiden eroista***

Taulukossa 3 on kuvattu ketterien ja suunnitelmapohjaisten menetelmien ominaispiirteitä neljästä eri näkökulmasta: kehitettävän sovelluksen, projektijohdon, teknisten ominaispiirteiden sekä henkilöstön ominaisuuksien näkökulmista.

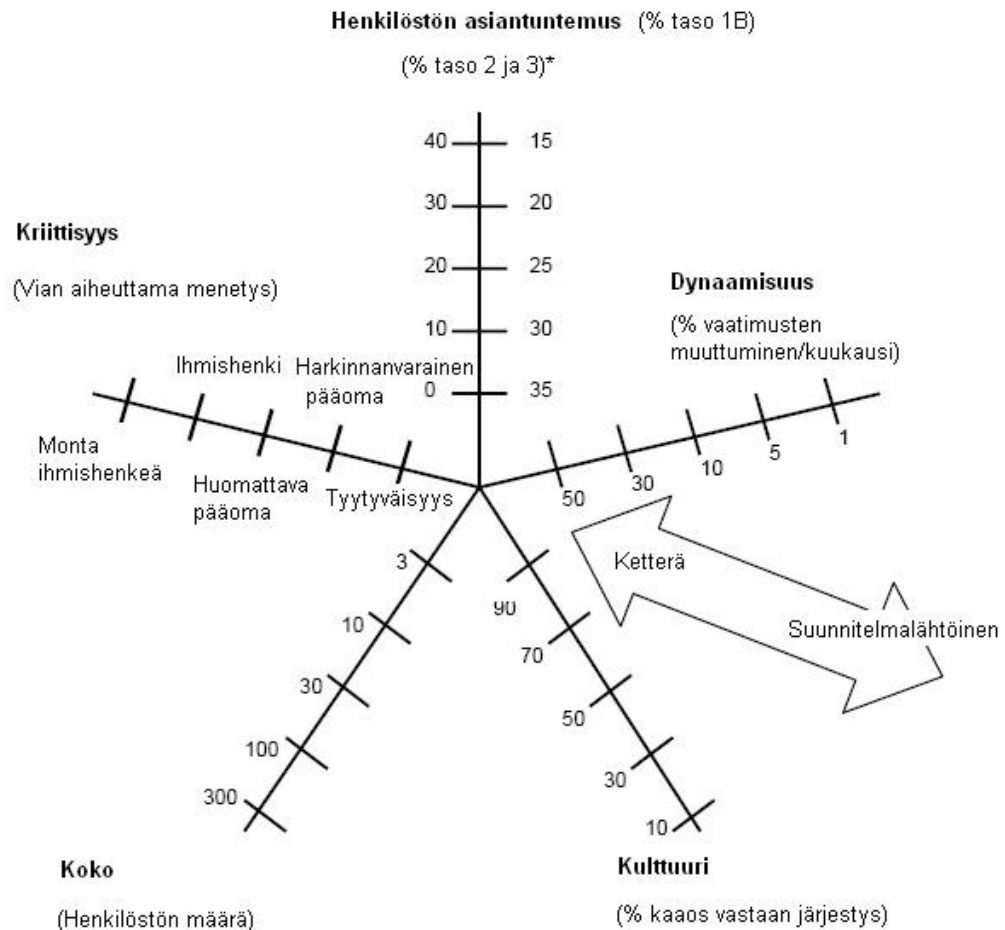


**Taulukko 3. Ketterien ja suunnitelmapohjaisten menetelmien ominaispiirteitä, muokattu lähteestä (Boehm ja Turner, 2004).**

Ominaispiirre	Ketterä	Suunnitelmapohjainen
<b>Kehitettävä sovellus</b>		
Päämäärät	Nopea lisäarvo; nopea vastaa- minen muutokseen.	Ennustettavuus, stabiilisuus.
Koko	Pienet tiimit ja projektit.	Suuret tiimit ja projektit.
Ympäristö	Turbulentti; paljon muutoksia.	Stabiili, vähän muutoksia.
<b>Projektijohto</b>		
Asiakassuhteet	Asiakkaita projektin tiloissa; keskittynyt tärkeimpien ominai- suuksien toteuttamiseen.	Asiakaskontakteja kun tarvi- taan; keskittynyt sopimusehto- jen täyttämiseen.
Suunnittelu ja kontrollointi	Sisäistetyt suunnitelmat, laadul- linen kontrolli.	Dokumentoidut suunnitelmat, tilastollinen kontrolli.
Kommunikaatio	Hiljainen (tacit) tietämys.	Dokumentoitu (codified) tietä- mys.
<b>Tekniset ominaispiirteet</b>		
Vaatimukset	Vaatimusten kehittymisen en- nustaminen mahdoton- ta/hankalaa.	Vaatimusten kehittymisen en- nustaminen mahdollista.
Kehitys	Yksinkertainen suunnittelu, pienet yhdellä kertaa toteutetta- vat ohjelmiston osat, ohjelmis- ton muokkaaminen oletettu hal- vaksi.	Laajamittainen suunnittelu, suu- ret yhdellä kertaa toteutettavat ohjelmiston osat, ohjelmiston muokkaus oletettu kalliiksi.
Testaus	Suoritettavat testitapaukset määrittävät vaatimukset.	Dokumentoidut testisuunnitel- mat ja -käytännöt.
<b>Henkilöstö</b>		
Asiakkaat	Tiiviissä tekemisissä projektis- sa.	Ei niin tiiviissä tekemisissä pro- jektin kanssa.
Kehittäjät	Suuremmat keskimääräiset osaamisvaatimukset kehittäjille.	Pienemmät keskimääräiset osaamisvaatimukset kehittäjille.
Kulttuuri	Löyhempi organisaatiokulttuuri.	Tiukat prosessit.

On myös mahdollista soveltaa ohjelmistokehitykseen niin sanottua hybridimallia, jossa suunnittelua vaativat osat suunnitellaan suunnitelmapohjaisilla menetelmillä ja soveltuvat osat ketterillä menetelmillä, (Boehm ja Turner, 2004). Tämä mahdollistaa

molempien menetelmien parhaiden puolien hyödyntämisen. Kuvassa 7 on esitetty dimensiot, jotka vaikuttavat käytettävän metodin valintaan. Hybridimallissa dimensioiden arviointi tehdään jaottelun perusteella kullekin toteutettavalle osalle.



\* Tasot 1B, 2 ja 3 kuvaavat Cockburnin ohjelmiston ymmärtämisen kolmea tasoa. Korkeammat tasot 2 ja 3 kuvaavat asiantuntijoita. Cockburnin asteikko on verrannollinen sovelluksen kompleksisuuteen. Kehittäjä voi olla tasolla 2 organisaatiossa, jossa kehitetään yksinkertaisia sovelluksia, mutta organisaatiossa, jossa kehitetään monimutkaisia sovelluksia sama kehittäjä voi olla tasolla 1A.

**Kuva 7. Metodien valintaan vaikuttavat dimensiot, alkuperäislähde (Boehm ja Turner, 2004), kuvan suomenkielinen käännös lähteestä (Kettunen, 2008).**

## 5 Ohjelmistotestaus

Testaukselle on useita eri määrittelyjä lähteestä riippuen. Haikalan ja Märijärven (Haikala ja Märijärvi, 2006) mukaan testaus määritellään perinteisesti: ”suunnitelmalliseksi virheiden etsimiseksi ohjelmaa tai sen osaa kuormittamalla”. Tässä työssä käytetään Edward Kitin määritelmää, jonka mukaan testaus koostuu sekä validoivasta että verifioivasta testauksesta (Kit, 1995). Testauksen lopullisena päämääränä on kuitenkin asiakkaiden tyytyväisyys, koska asiakkaat maksavat viimekädessä ohjelmistokehityksen (Kit, 1995).

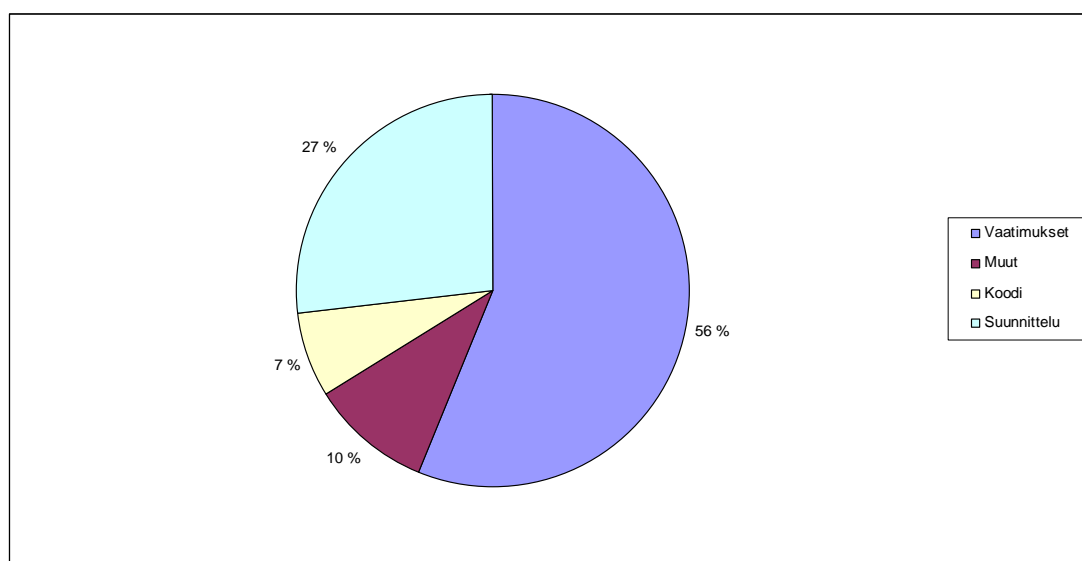
Verifioivalla testauksella pyritään varmistamaan, että asiakkaan vaatimukset ovat oikein määriteltynä, ja asiakasvaatimukset siirtyvät oikein muihin suunnitteludokumentteihin, jotta ohjelmistotuotantoprosessissa tehdään oikeaa tuotetta. Validoivalla testauksella taas pyritään varmistamaan, että tuotetta tehdään oikein vertaamalla suunnitteludokumenttien määrittelyä toteutettuun osaan.

Tärkeänä osana virheiden löytämisessä on se, että tuotoksen testaa joku muu henkilö kuin tuotoksen tekijä. Äkkiseltään ajateltuna tuntuisi loogiselta, että tekijä itse testaa tuotoksensa, koska hän sen parhaiten tuntee ja näin ollen pystyy testaamaan sen tehokkaimmin. Kehittäjien tavoitteena kuitenkin on saada aikaan valmista ohjelmakoodia mahdollisimman nopeasti, mikä on täysin vastakkainen testaajan tavoitteelle, joka on löytää mahdollisimman paljon virheitä. Mikäli testaamme itse tekemäämme materiaalia, niin tiedämme missä mahdolliset rajoitteet ja ongelmat ovat ja ajatellaan, että ne eivät ole suuria ongelmia, vaikka ne todellisuudessa voivat sitä olla (Kit, 1995).

Ohjelmistokehityksajasta ja -resursseista 40-70% kuluu virheiden havaitsemiseen ja poistoon, (Kit, 1995). Näin ollen testauksella on erittäin merkittävä rooli ohjelmistokehityksen kokonaiskustannuksista ja potentiaalia suuriin kustannussäästöihin.

Kuvassa 8 on esitetty virheiden jakauma suunnitelmapohjaista menetelmää käytettäessä eri kehitysprojektin vaiheiden välillä. Kuvasta voidaan havaita, että suurin osa

virheistä syntyy jo vaatimusmäärittelyn aikana. Erityisen huomioitavaa kuvassa on, että vain 7 % virheistä syntyy koodauksessa. Loogisena johtopäätöksenä tästä on, että suurin osa virheistä on mahdollista poistaa jo vaatimusmäärittelyn aikana. Lisäpontta virheiden mahdollisimman varhaiselle poistamiselle antaa se, että mikäli esimerkiksi vaatimusmäärittelyssä syntynyt virhe pääsee siirtymään myöhempisiin vaiheisiin, on sen korjaaminen silloin huomattavasti kalliimpaa.

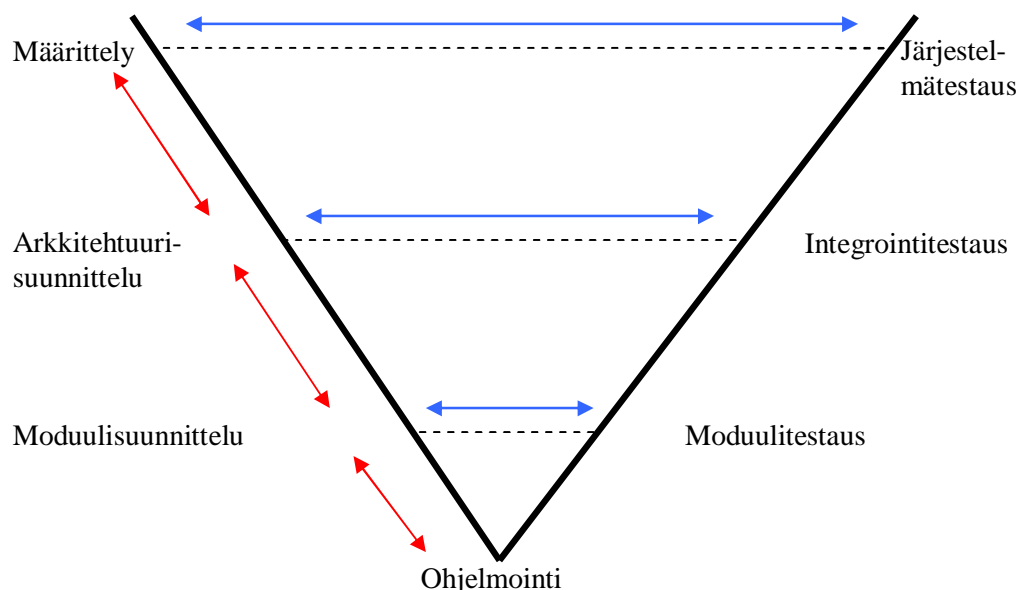


**Kuva 8. Virheiden jakauma, (Bender, 1993)**

Esimerkiksi mikäli vaatimusmäärittely on virheellinen, tehdään järjestelmän arkkitehtuurisuunnittelu virheelliseen dokumenttiin perustuen, jolloin lopputuloksena on vääränlainen arkkitehtuuri. Jo tässä vaiheessa virheen korjaaminen on huomattavasti kalliimpaa kuin määrittelyvaiheessa, koska muutokset joudutaan tekemään kahteen dokumenttiin ja työ toistamaan. Mikäli vielä tässäkin vaiheessa ei virheitä havaita ja järjestelmää ryhdytään toteuttamaan, on virheiden korjaaminen jo todella kallista, koska korjaus pitää tehdä vaatimusmäärittelyyn, tekniseen määrittelyyn, koodiin ja testitapauksiin.

## 5.1 Testaus suunnitelmapohjaista ohjelmistokehitysmenetelmää käytettäessä

Suunnitelmapohjaista ohjelmistokehitysmenetelmää käytettäessä validoivaa ja verifioivaa testausta havainnollistaa hyvin testauksen V-malli (kuva 9).



**Kuva 9. Ohjelmistotestauksen V-malli, muokattuna lähteestä (Haikala ja Märijärvi, 2006)**

Kuvassa aika kulkee vasemmalta oikealle. Tästä tulee V-mallin V-muoto. Aikajanaa ei kuitenkaan ole piirretty kuvaan sen selkeyden säilyttämiseksi. V-malli liittyy läheisesti vesiputousmalliin, joka esitettiin kappaleessa 4.1.

Kuvaan tummalla merkityt nuolet tarkoittavat verifioivaa testausta. Esimerkiksi määrittelyn ja arkkitehtuurisuunnittelun välillä se tarkoittaa, että määrittelydokumentaatiosta tiedot ovat siirtyneet oikein arkkitehtuurisuunnittelu-dokumentaatioon eli tiedot on verifioitu. Käytännössä siinä ei siis varsinaisesti ajeta kehitettävää ohjelmaa vaan verifiointi suoritetaan dokumentti-tasolla.

Vaalealla merkityt nuolet puolestaan tarkoittavat validoivaa testausta. Validoivassa testauksessa ohjelmoitua osaa tai osakokonaisuutta verrataan vastaavan tason doku-

mentteihin. Esimerkiksi moduulitestauksessa koodattua moduulia verrataan siihen, mitä moduulisuunnitteludokumentissa on määritelty. Moduulitestaukselle synonyymina on tässä työssä käytetty sanaa yksikkötestaus.

Seuraavissa kappaleissa kerrotaan, mitä validoiva ja verifioiva testaus tarkemmalla tasolla tarkoittavat ja mitä testauksen alatyyppejä ne sisältävät.

### **5.1.1 Verifioiva testaus**

Verifioivalla testauksella tarkoitetaan ohjelmistokehityksen eri vaiheissa tuotetulle dokumentaatiolle tehtävää tarkastusta. Kitin (Kit, 1995) määritelmän mukaan verifiointi on ihmisen tekemää tarkastusta tai katselmointia työn tuotokselle.

Esimerkiksi vaatimusmäärittely-, toiminnallista määrittely-, teknistä määrittelydokumentaatiota, koodia tai testitapauksia voidaan verifioida. Kitin (Kit 1995) mukaan verifiointimenetelmiä on kolmenlaisia:

- 1) Vertaistarkastukset (peer-review)
- 2) Läpikäynnit (walkthroughs)
- 3) Tarkastukset, katselmoinnit (inspections)

Vertaistarkastus on verifiointimenetelmistä kevyin. Siinä joku muu kuin dokumentin tekijä käy dokumentin läpi ja merkitsee havaitsemansa virheet.

Läpikäynnit ovat virheiden löytämisen näkökulmasta vertaistarkastusta tehokkaampi menetelmä. Siinä useampi henkilö kokoontuu ja tarkastettava lopputuote käydään läpi esittelijän johdolla (yleensä tarkastettavan dokumentin tekijä). Huomattava on, että henkilöt tulevat tilaisuuteen valmistautumatta.

**Taulukko 4. Verifiointimenetelmien perusominaisuudet, (Kit, 1995)**

	<b>Tarkastukset</b>	<b>Läpikäynnit</b>	<b>Vertais-tarkastukset</b>
Esittelijä	Ei dokumentin tekijä.	Kuka tahansa (yleensä dokumentin tekijä).	Ei kukaan.
Osallistujat	3-6 henkilön tiimi.	Useita henkilöitä.	1 tai 2.
Valmistelu	Kyllä.	Ainoastaan esittelijä valmistelee.	Ei.
Virheiden luokitus	Kyllä.	Ei vaadittu.	Ei.
Raportti	Kyllä.	Ei vaadittu.	Verbaalinen kommunikointi.
Edut	Tehokas menetelmä.	Osallistuvat tutustuvat hyvin läpikäytävään kohteeseen.	Halpa tehdä.
Haitat	Lyhyellä aikavälillä mitattuna kallis.	Löytää tarkastuksiin verrattuna vähemmän virheitä.	Löytää sekä läpikäynteihin että tarkastuksiin nähden vähemmän virheitä.

Tarkastukset ovat näistä kolmesta mainitusta menetelmästä kaikkein muodollisin, mutta myös uusien virheiden havaitsemisen kannalta tehokkain. Tarkastukset eroavat läpikäynnistä siinä, että ennen tilaisuuteen tuloa siihen osallistuvat henkilöt ovat valmistautuneet tilaisuuteen lukemalla tarkastettavat dokumentit läpi. Tilaisuudessa dokumentin esittelijänä on muu henkilö kuin dokumentin tekijä. Tilaisuuden lopputuloksena tuotetaan kirjallinen raportti ja löydetty virheet luokitellaan ja lasketaan luokittain, joiden perusteella voidaan esimerkiksi arvioida kuinka tehokasta tarkastustoiminta oli. Taulukkoon 4 on koottu verifiointimenetelmien perusominaisuudet.

Tärkeä työkalu verifioimisessa ovat tarkastuslistat, joissa on määritelty dokumentista tarkastettavat asiat. Yleisesti ottaen tarkastettaville dokumentaatiotyypeille on määritelty omat tarkastuslistansa tarkastettavista asioista. Esimerkiksi testitapausten tarkistuslista voisi sisältää tarkastettavina kohtina esimerkiksi seuraavat asiat (lista on esimerkinomainen, todellisuudessa lista olisi huomattavasti kattavampi):

- 1) Tarkista, että kentiin tietoa syötettäessä on huomioitu skandinaaviset kirjaimet.

- 2) Tarkista, että kenttiin tietoa syötettäessä on huomioitu, että myös numeroita voi syöttää.
- 3) Tarkista, että kentän maksimipituus ja minimipituus on testattu.
- 4) Tarkista, että uuteen ruutuun siirryttäessä ruudun ulkoasu on validoitu.
- 5) Tarkista, että kaikkien linkkien toiminta jokaisella ruudulla on testattu.

Tarkistuslistat ovat tehokas keino virheiden poistamiseen edellyttäen, että ne ovat kattavia ja sisältävät sellaiset tarkasteltavat kohdat, jotka tyypillisesti unohdetaan määrittellä.

### 5.1.2 Validoiva testaus

IAEE/ANSI:n määritelmä validoinnille: ”Validation is the process of evaluating a system or component during or at the end of the development process to determine whether *it satisfied specified requirements*”, (IEEE/ANSI, 1983). Validoinnissa siis verrataan dokumentaatioissa kirjallisesti kuvattua määrittelyä siihen, miten komponentti ja/tai ohjelma on toteutettu.

Validoivaan testaukseen keskeisenä käsitteenä liittyy testitapaus, joka IEEE/ANSI:n määritelmän mukaan tarkoittaa seuraavaa: ”A set of test inputs, execution conditions, and expected results developed for a particular objective.” ja ”The smallest entity that is always executed as a unit, from beginning to end.”, (IEEE/ANSI, 1983). Testitapaus on perusyksikkö, jolla järjestelmän yhtä ominaisuutta tai useampia ominaisuuksia testataan. Testitapaukseen läheisesti liittyy testiskripti, joka on testitapauksen suorittamiseen tarvittava ”käyttöohje”. Testiskriptin pitäisi olla niin yksityiskohtaisesti kirjoitettu, että periaatteessa kuka tahansa sen perusteella pystyisi suorittamaan siihen liittyvän testitapauksen.

Validoiva testaus voidaan jakaa musta- ja lasilaatikko testaukseen. Mustalaatikko testauksessa testaja ei tunne ohjelman sisäistä rakennetta kun taas lasilaatikkotestauksessa sisäinen rakenne tunnetaan. Kitin mukaan nämä ovat fundamentaalisia testausstrategioita, (Kit, 1995).



Mustalaatikkotestausta käytetään silloin, kun testitapaukset tehdään toiminnallisen määrittelyn perusteella. V-mallissa tämä tapahtuu järjestelmätestauksen tasolla. Vastaavasti lasilaatikkotestausta käytetään, kun testitapaukset johdetaan teknisen määrittelyn perusteella. V-mallissa tämä tapahtuu järjestelmätestauksen alapuolisilla tasoilla.

Mustalaatikkotestaukseen liittyvä sisäisen rakenteen tuntemattomuus on hyödyllistä, koska ohjelmakoodin näkemisen jälkeen vaatimukset nähdään eri tavalla. Mustalaatikkotestauksessa vaatimusten tulkinta tehdään loppukäyttäjän näkökulmasta. Mikäli testaaja näkee ohjelmakoodin, se saattaa muuttaa hänen tekemäänsä tulkintaa vaatimuksista ja vaikeuttaa asettumista loppukäyttäjän asemaan (Kit, 1995).

On tärkeää huomata, että mustalaatikkotestaus ei testaa piilossa olevia toimintoja, eli toimintoja jotka on toteutettu, mutta joita ei ole kuvattu toiminnallisessa määrittelyssä. Näin ollen menetelmän avulla ei voi löytää niihin liittyviä virheitä. Vastaavasti lasilaatikkotestauksella ei havaita puuttuvia toimintoja, jotka on kuvattu toiminnallisessa määrittelyssä, mutta joita ei ole kuvattu/toteutettu teknisessä suunnitellussa tai koodissa, (Kit, 1995). Ongelmat pystytään korjaamaan verifioivalla testauksella, joilla dokumentit saadaan yhdenmukaiseksi sekä validoimalla jolloin tarkastetaan, että toteutus vastaa määrittelyä. Molempien tyyppisiä testauksia siis tarvitaan.

Validoiva testaus voidaan jakaa eri osiin testattavan tason mukaan:

- 1) Matalan tason testaus
  - a. Yksikkö (moduuli) testaus
  - b. Integraatiotestaus
- 2) Korkean tason testaus
  - a. Käytettävyys testaus
  - b. Toiminnallinen testaus
  - c. Systemitestaus
  - d. Hyväksymistestaus

Matalan tason testauksessa testaus on lasilaatikkotestausta, kun taas korkean tason testauksessa mustalaatikkotestausta. Korkean tason testauksessa abstraktiotaso on suurempi kuin matalan tason testauksessa ja siitä on peräisin myös testausmenetelmän nimi.

Yksikkötestauksella tarkoitetaan pienimmän toteutettavan yksikön – yleensä moduulin testausta. Integraatiotestauksella testataan, kuinka moduulit toimivat yhteen integroituna. Yksikkö- ja integraatiotestauksen suorittavat käytännössä aina järjestelmän kehittäjät. Tämä johtuu siitä, että sopivien matalan tason testien tekeminen vaatii moduulien yksityiskohtaista tuntemista, jonka hankkiminen ulkopuoliselle testaajalle tulisi kalliiksi. Sen sijaan yksikkö-/integraatiotestit voi tehdä eri ohjelmoija kuin henkilö, joka komponentit on ohjelmoinut.

Yksikkötestaukseen liittyy kiinteästi koodikattavuus (code coverage), joka tarkoittaa määrää, kuinka paljon ohjelmakoodista on testattu ja se ilmoitetaan prosenttilukuna. Koodikattavuutta voidaan mitata useilla eri kriteereillä, esimerkiksi (Kit, 1995):

- Ehtokattavuutta (condition coverage) – mittaa kuinka paljon ehtojen mahdollisista arvoista on testattu.
- Polkukattavuutta (path coverage) – mittaa kuinka moni mahdollisista ohjelmapoluista on testattu.
- Lausekattavuutta (statement coverage) – mittaa kuinka moni ohjelman koodiriveistä on testattu.
- Päätöskattavuutta (branch coverage) – mittaa kuinka moni päätöksistä (esimerkiksi if-lauseet) on testattu.

Mitä lähempänä lukema on 100 % sitä parempi kattavuus. Kriteerit eivät suoraan ole keskenään vertailukelpoisia. Esimerkiksi 80 % polkukattavuus testaa ohjelmistosta suuremman osan kuin 80 % lause- tai päätöskattavuus.

Korkean tason testauksista käytettävyydestestauksella testataan nimensä mukaisesti ohjelman käytettävyyttä. Se voidaan toteuttaa esimerkiksi tarkkailemalla, kuinka käyttäjät käyttävät ohjelmaa ja sen perusteella tekemään mahdolliset tarvittavat muutok-

set käyttöliittymään. Käytettävyydestäuksen toteuttamiseen on useita eri tekniikoita, joiden tarkempi kuvaaminen on diplomityön aihealueen ulkopuolella.

Toiminnallisessa testauksessa testataan järjestelmää vertaamalla toiminnallista määrittelydokumentaatiota toteutukseen. Mikäli havaitaan poikkeama, on virhe joko toteutuksessa tai määrittelyssä ja jompaakumpaa korjataan. Mikäli verifioiva testaus on tehty täsmälleen oikein, eli toiminnallinen määrittelydokumentaatio on virheetön, on virhe toteutuksessa. Virheettömien dokumenttien teko on käytännössä mahdotonta, koska ei ole mahdollista ottaa huomioon kaikkea mahdollista dokumenttia tehdessä. Iso osa virheistä, jotka olisi pystytty löytämään verifioivalla testauksella, löytyvätkin vasta toiminnallisessa testauksessa. Sekä validoivaa että verifioivaa testausta siis tarvitaan.

Käytännön kokemukseni perusteella sekä käytettävyy- että toiminnallinen testaus mielletään yleensä kuuluvaksi systeemitestauksen alatyypeiksi, mutta Kitin (Kit, 1995) mukaisessa jaottelussa ne ovat systeemitestauksen rinnalla.

Systeemitestaus koostuu seuraavista alatyypeistä, (Kit, 1995)

- Volyymitestaus
- Kuormitustestaus
- Turvallisuustestaus
- Suorituskykytestaus
- Resurssien kulutustestaus
- Konfiguraatiotestaus
- Yhteensopivuustestaus
- Asentuvuustestaus
- Palautuvuustestaus
- Luotettavuustestaus

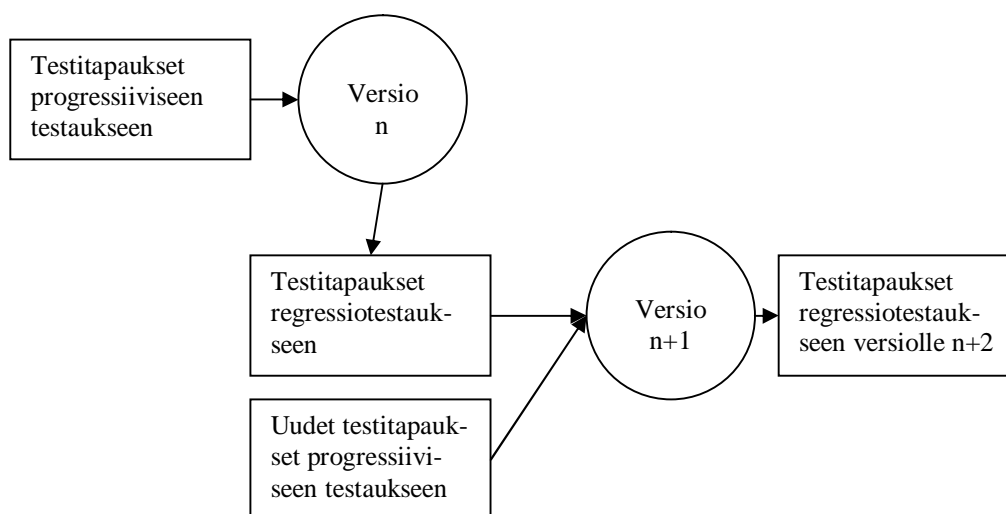
Systeemitestauksen alatyypien nimistä voi päätellä, mitä niillä testaan. Oleellista niiden testaamisessa on, että niitä testataan asiakasvaatimuksia vasten. Esimerkiksi yhteensopivuustestauksessa testataan, täyttääkö järjestelmä yhteensopivuusvaatimuk-

set esimerkiksi yrityksen nykyisten järjestelmien kanssa. Tämän tyyppisiä asiakasvaatimuksia sanotaan ei-toiminnallisiksi vaatimuksiksi.

Hyväksymistestaus on yleensä asiakkaan suorittamaa testausta järjestelmän lopullisessa käyttöympäristössä. Systemitestaukseen verrattuna erona on testauksen suorittajan lisäksi se, että hyväksymistestaus on yleensä huomattavasti suppeampaa. Käytännön kokemuksiini perustuen esimerkiksi suorituskykytestejä ei hyväksymistestauksessa tehdä. Tämä johtuu siitä, että ne pitäisi olla tehty (mikäli tarvetta) systemitestausvaiheessa ja niiden tekeminen vaatii erityisasiantuntemusta, jota ei asiakkaalla välttämättä ole. Hyväksymistestaus tehdään asiakasvaatimuksiin nähden ja V-mallissa se sijoittuu samalle tasolle kuin systemitestaus.

Validoivaan testaukseen liittyy myös termi regressiotestaus. Regressiotestauksella pyritään varmistamaan, että ohjelmaan lisätyt toiminnot/muutokset eivät ole rikkoneet aikaisemmin toteutettuja toimintoja. Rikkoutuminen on hyvin yleistä, sillä ohjelmistoissa on paljon riippuvuussuhteita eri ohjelmiston osiin ja toteuttaja ei välttämättä ymmärrä muutoksia tehdessään mihin kaikkialle ne vaikuttavat.

Kun uusia ominaisuuksia testataan ensimmäistä kertaa, sanotaan tätä progressiiviseksi testaukseksi. Kun progressiivista testausta varten tehdyt testitapaukset on saatu suoritettua hyväksytysti (eli toteutus vastaa määrittelyjä), niistä muodostetaan testit regressiotestausta varten. Kuva 10 havainnollistaa tätä.



**Kuva 10. Regressiotestitapausten muodostuminen.**

Testitapausten suorittamista voidaan myös automatisoida sopivilla työkaluilla. Eri-tyisesti regressiotestitapaukset pyritään automatisoimaan, koska niiden tarkoituksena on testata, ettei mitään ole rikkoutunut. Ongelmana testauksen automatisoinnissa on testitapausten uudelleenkäytettävyys (uuteen projektiin joudutaan tekemään kokonaan uudet testitapaukset) sekä joissakin tapauksissa korkea hinta. Korkea hinta johtuu esimerkiksi uudelleenkäyttämättömyydestä sekä vaaditusta asiantuntijuudesta (henkilöstökuluista), joka vaaditaan automaatiotyökalujen käyttöön.

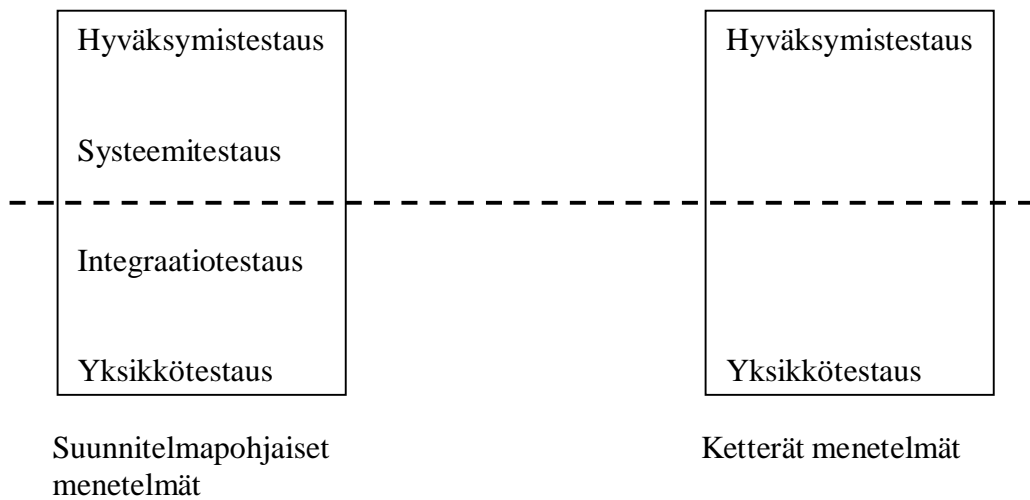
Automatisoinnin yksityiskohtaisempi käsittely on tämän diplomityön aihealueen ulkopuolella.

## ***5.2 Testaus ketterää ohjelmistokehitysmenetelmää käytettäessä***

Ketteriä ohjelmistokehitysmenetelmiä käytettäessä tapahtuu vain kahdenlaista testausta: yksikkötestausta ja hyväksymistestausta. Lisäksi V-mallin mukaista verifioivaa testausta dokumentoimalla ei ketterää menetelmää puhtaasti sovellettaessa tapahdu lainkaan, koska ketterien menetelmien yksi perusajatus on dokumentoida niin vähän kuin mahdollista.

Ketterää menetelmää käytettäessä tapahtuu kuitenkin myös verifiointia kun hiljaista tietoa siirtyy esimerkiksi palavereissa. Perusteluina tälle väittämälle on se, että ketterää menetelmää sovellettaessa ei voida paeta kahta ohjelmistokehityksen peruskysymystä: Teemmekö oikeaa tuotetta? (validoiva testaus) ja Teemmekö tuotetta oikein? (verifioiva testaus). Näin ollen myös ketterässä testauksessa molemmat testauksen peruskomponentit ovat läsnä.

Kuvassa 11 on esitetty vertailun vuoksi testaus molempia menetelmiä käytettäessä.



**Kuva 11. Testauksen tasot eri ohjelmistokehitysmenetelmiä käytettäessä (Wikström, 2007)**

Testaajan rooli ketterää menetelmää käyttäessä on erilainen suhteessa suunnitelmapohjaisiin menetelmiin käytettäessä. Koska dokumentaatio pyritään pitämään mahdollisimman pienenä, ei hyväksymistestejä voida kirjoittaa dokumentaatioon perustuen. Samasta syystä järjestelmän toimintaa ei täysin pysty selvittämään dokumentaation perusteella, vaan toiminta selviää esimerkiksi päivittäisissä palavereissa ja keskustelemalla ohjelmistokehittäjien kanssa. Yleensä hyväksymistestit tehdään yhteistyössä asiakkaan kanssa.

Ketteriä menetelmiä käytettäessä myös testaajalta vaaditaan päivittäistä kommunikointia kehittäjäryhmän kanssa huomattavasti enemmän kuin suunnitelmapohjaista menetelmää käytettäessä. Kommunikointia helpottamiseksi on hyvä sijoittaa kehittäjät ja testaajat samaan tilaan mikäli mahdollista. Testausta tapahtuu jatkuvasti projektin edetessä, koska ketteriä menetelmiä käytettäessä toteutetut uudet osat liitetään ohjelmaan välittömästi, jolloin ne ovat testattavissa. Tämä tarkoittaa jatkuvaa integrointia, (continuous integration). Suunnitelmapohjaisia menetelmiä käytettäessä testattavaa tulee kerralla yleensä huomattavan paljon. Testaajat antavat jatkuvaa palautetta järjestelmän toimivuudesta kun taas suunnitelmapohjaisissa kaikki palaute tulee vasta lopuksi. (Hendrickson, 2008).

Suunnitelmapohjaisia menetelmiä käytettäessä testaus suunnitellaan ja niin sanotut entry ja exit –kriteerit määritellään ennakkoon. Entry-kriteereillä tarkoitetaan sitä, että testattavan ohjelman on täytettävä tietyt vaatimukset, kun se otetaan testattavaksi. Exit-kriteerillä tarkoitetaan tiettyä vaatimusta esimerkiksi läpäistyjen testitapausten suhteen, jotta testaus voidaan lopettaa. Ketteriä menetelmiä käytettäessä suunnittelua tai kriteerien määrittelyä ei tehdä ennakkoon, vaan ne päätetään projektin aikana muun kehittäjäryhmän kanssa, koska myös testauksessa pyritään minimoimaan dokumentaatiota. Suunnitelmapohjaisia menetelmiä käytettäessä järjestelmän laatuvas- tuu on enemmän testaajilla, kun taas ketterissä menetelmissä järjestelmän hyväksy- minen tai hylkääminen tehdään tiimin yhteisenä päätöksenä. (Hendrickson, 2005)

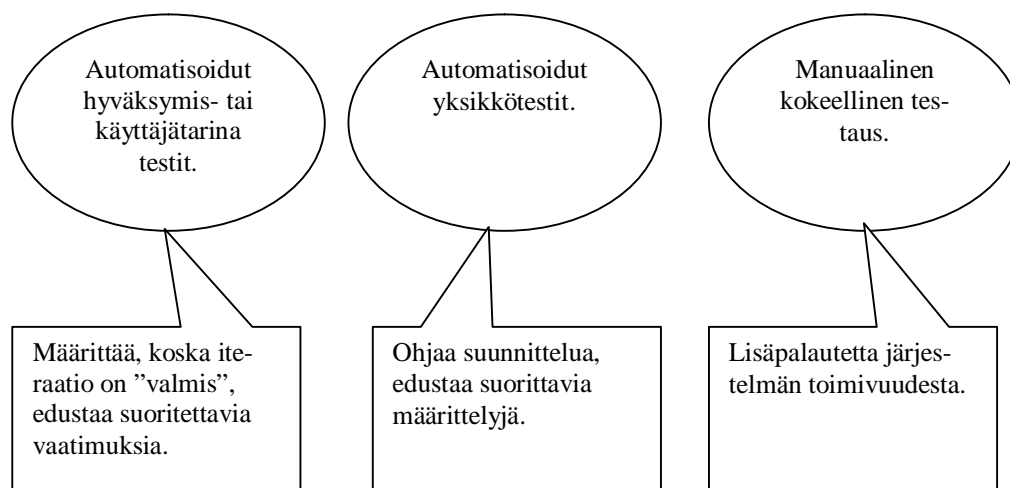
Ketterissä menetelmissä myös kokeellisella testauksella (exploratory testing) on mer- kittävä osuus. James Bachin ytimekkään määritelmän mukaan kokeellinen testaus tarkoittaa samanaikaista oppimista, testien suunnittelua ja testien suorittamista, (Bach, 2003). Tällä tarkoitetaan toisin sanoen sitä, että testejä suoritettaessa välitty- vää informaatiota käytetään hyväksi jo olemassa olevien testien muokkaamiseksi ja uusien testien luomiseksi. Lisäksi virheitä pyritään löytämään käyttämällä järjestel- mää ja testaamalla potentiaalisia virhepaikkoja. Tämä asettaa testaajien osaamiselle kovemmat vaatimukset, sillä virheitä on todella etsittävä ja tiedettävä, mistä virheitä voi löytyä (Santamaria, 2007).

Kokeellinen testaus voidaan ajatella sijaitsevan jatkumon toisessa ääripäässä, jossa testauksen idea keksitään testausta suoritettaessa. Toisessa ääripäässä on täysin skrip- tatut testitapaukset, jotka tehdään puhtaasti määrittelydokumentaation perusteella ja joita ei testauksen suorituksen tuottaman informaation perusteella muokata. Yleensä yrityksissä on käytössä kokeellista testausta myös suunnitelmapohjaista menetelmää käytettäessä ainakin jossain määrin (Bach, 2003). Tällöin kuitenkin ollaan lähempänä täysin skriptattua ääripäätä. Ketterää menetelmää sovellettaessa ollaan lähellä jatku- mon kokeellista ääripäätä.

Kuten suunnitelmapohjaisia menetelmiä käytettäessä, myös ketterissä menetelmissä, yksikkötestit kirjoittavat ohjelmoijat. Jos ketterää menetelmää sovellettaessa yksik-

kötestit kirjoitetaan ennen varsinaista ohjelmakoodia, puhutaan TDD:stä (Test Driven Development). Tällöin esimerkiksi pääsuunnittelija kirjoittaa testit ennakkoon, jolloin yksittäisen ohjelmoijan on toteutettava komponentti siten, että testi menee läpi.

Kuva 12 esittää testausta iteraation sisällä. Joissakin organisaatioissa hyväksymistestit kirjoitetaan ensimmäiseksi ennen kuin varsinaista ohjelmointia on edes aloitettu. Ne pyritään yleensä automatisoimaan ja niistä tehdään regressiotestejä samoin kuin suunnitelmapohjaista menetelmää käytettäessä. Tällöin herää kysymys, mihin yksikkötestejä tarvitaan? Niitä tarvitaan siihen, että ne testaavat asioita huomattavasti tarkemmalla tasolla kuin hyväksymistestit, jotka testaavat vain yleistä toiminnallisuutta. Yksikkötestejä on myös huomattavasti enemmän kuin hyväksymistestejä.



**Kuva 12. Ketterien menetelmien iteraatioissa tapahtuva testaus, (Hendrickson, 2005)**



## **6 MUITA OHJELMISTON LAATUUN VAIKUTTAVIA TEKIJÖITÄ**

Seuraavissa kappaleissa käsitellään välillisesti ohjelmiston laatuun vaikuttavien hypoteesien teoreettinen tausta.

### **6.1 Standardin toteutuminen**

Tämän diplomityön lähtökohtana ovat testausstandardi ISO/IEC29119 ja laatustandardi ISO/IEC 25010 (ISO/IEC, 2008). Molemmat standardit olivat diplomityön teon aikaan keskeneräisiä, mutta tutkimusryhmällä oli mahdollisuus käyttää molempia ja näin ollen käytettävissä oli uusinta mahdollista tietoa. Tässä tutkimuksessa testaus ja laatu ymmärretään näiden standardien kautta ja oletetaan, että näiden standardien noudattaminen tuottaa laatua. Hypoteesi ”Standardin toteutuminen” perustuu tähän olettamukseen. Sen mukaan mitä paremmin standardi toteutuu organisaatioyksikössä, sitä parempi on ohjelmiston laatu.

### **6.2 Asiakkaan osallistuminen**

Asiakkaan osallistumisella ja asiakastyytyväisyyden välillä on tutkimusten mukaan todettu olevan positiivinen korrelaatio. Asiakkaan osallistumisella tarkoitetaan esimerkiksi asiakasyrityksen johdon osallistumista suunnitteluun ja suunnitteludokumenttien hyväksymiseen sekä järjestelmän varsinaisten käyttäjien osallistumista suunnitteluun. Aktiivinen osallistuminen parantaa kehitettävän järjestelmän määrittelyä ja tätä kautta edesauttaa sen suunnittelua. (Guimaraes et al., 1994)

On myös havaittu, että käyttäjät jotka osallistuvat järjestelmän suunnitteluun arvostavat järjestelmää enemmän verrattuna käyttäjiin, jotka eivät osallistuneet. Vastaavasti käyttäjät jotka eivät osallistuneet järjestelmän suunnitteluun todennäköisemmin vastustavat järjestelmän käyttöönottoa. (Guimaraes et al., 1994)

Asiakastyytyväisyys tarkoittaa Ishikawan määritelmän mukaan laatua (katso kappale 3.1). Tästä saamme teoreettisen perustelun tutkimuksen hypoteesille, jonka mukaan mitä suurempi asiakkaan osallistuminen, sitä parempi on ohjelmiston laatu.

Asiakkaan osallistumista mitattiin Olsonin and Ivesin kehittämällä modifioidulla lomakkeella (Olson ja Ives, 1980), jota käytettiin myös Guimaraesin (Guimaraes et al., 1994) tutkimuksessa.

### **6.3 Asiakassuhteen vaikutus laatuun**

Asiakassuhteella on tutkimusten mukaan todettu olevan vaikutusta ohjelmiston laatuun. Pasi, Smolanderin ja Nikulan (Nikula et al., 2008) mukaan ongelmat johtuvat asiakkaan määräämästä ohjelmistokehitysmenetelmän valinnasta ja valta-asymmetriasta. Valta-asymmetria tarkoittaa vallan epätasaista jakautumista asiakkaan ja toimittajan välillä.

Tässä tutkimuksessa mitataan valta-asymmetrian vaikutusta laatuun. Asiakkaan ollessa toimittajaa huomattavasti heikompi on sillä mahdollisesti laatuun negatiivisia seurauksia. Tämä johtuu siitä, että heikko asiakas suostuu toimittajan pyyntöihin asiakkaan menettämisen pelossa. Tällöin asiakas saattaa suostua asiakkaan muutospyyntöihin aivan projektin loppumetreillä, jolla saattaa olla vaikutusta aikatauluun ja ohjelmiston huonoon laatuun (Nikula et al., 2008). Tutkimuksen hypoteesin mukaan mitä pienempi ero valta-asymmetriassa, sitä parempi on ohjelmiston laatu.

### **6.4 Luottamus**

Luottamukselle on olemassa useita eri määritelmiä (Oza, 2006). Tässä työssä käytetään Sabherwalin määrittelemää luottamusta, joka on: “confidence that the behavior of another will conform to one’s expectations and in the goodwill of another” (Sabherwal, 1999).

Luottamus on erityisen tärkeää silloin, kun toimittaja ei ole asiakkaan kanssa läheisessä yhteistyössä eli ohjelmistokehitys on esimerkiksi ulkoistettu. Luottamuksen puute saattaa aiheuttaa liiallista kontrollointia esimerkiksi sopimusteitse ja asiakkaan jatkuvat tarkkailukäynnit toimittajan luona oikean tekemisen todentamiseksi voivat vähentää laatua. Vastaavasti liiallinen luottamus asiakkaaseen voi tuottaa ikäviä yllätyksiä, mikäli asiakkaalla ei ole kykyä tehdä sitä mitä on luvannut. Näin ollen tulisi löytää sopiva tasapaino kontrollon ja luottamuksen välille (Sabherwal, 1999).

Sabherwalin tutkimuksessa havaittiin neljää eri luottamustyyppiä:

- 1) Laskennallisperusteinen (calculus based) liittyy projektikohtaisiin palkkioihin ja rangaistuksiin, jotka on määritelty sopimusteitse. Sopimukseen sisällytettävillä rangaistuspykälillä tai muulla kontrolloinnilla, kuten raporttien vaatimisella ja sopimalla muutoshallintakäytännöistä minimoidaan toimittajan mahdollinen opportunistinen käyttäytyminen. Tämän tyyppinen luottamus voi myös perustua toimittajan oletukseen siitä, että jatkossa saadaan uusia projekteja.
- 2) Tietämysperusteinen (knowledge based) pohjautuu siihen kuinka hyvin osapuolet tuntevat toisensa. Tämä voi pohjautua siihen, että toimittaja on jo aikaisemmin toimittanut järjestelmän asiakkaalle tai että asiakas pyytää toimittajalta ensin pienen järjestelmän, jotta asiakas vakuuttuu toimittajasta.
- 3) Identifiointiperusteinen (identification based) perustuu osapuolten toteamaan yhteiseen päämäärään ja sen korostamiseen. Yhteinen päämäärä saavutetaan selkeällä ja riittävällä kommunikoinnilla projektin molempien osapuolten kesken.
- 4) Suoritusperusteinen (performance based) perustuu projektin päämäärien saavuttamiseen. Päämäärien saavuttaminen kasvattaa luottamusta osapuolten välillä.

Tutkimuksen hypoteesin mukaan parempi luottamus parantaa laatua ja vastaavasti matalampi luottamus vähentää sitä. Tutkimuksessa oletetaan, että kontrollon ja luottamuksen tasapaino pysyy sopivana ja näin ollen korkeampi luottamus tarkoittaa korkeampaa laatua.

Luottamusta mitattiin Bentonin ja Malonin kehittämillä kysymyksillä luottamuksen mittaamiseksi (Benton ja Maloni, 2004).

## **6.5 Kriittisyys**

Ohjelman kriittisyydellä tarkoitetaan vahingon määrää, joka ohjelman virheestä voi aiheutua. Pienimmillään ohjelman virhe aiheuttaa lähinnä tyytymättömyyttä kun taas suurimmillaan vaikutuksena voi olla ihmishenkien menetyksiä. Esimerkiksi todella kriittisiä sovelluksia ovat ydinvoimaloissa, auton turvatyynyissä ja avaruussukkulassa tarvittavat ohjelmat. Toisessa ääripäässä ovat esimerkiksi yksinkertaiset peliohjelmat. Ohjelmavirheen suuremmasta haitasta johtuen kriittiseen ohjelmistoon ei saa jäädä virheitä niin paljon kuin vähemmän kriittiseen. Tutkimuksen hypoteesin mukaan vähäisempi virhemäärä tarkoittaa ohjelmiston parempaa laatua ja näin ollen mitä kriittisempi ohjelma sitä parempi laatu.

Kriittisten sovellusten kehittäminen vaatii suunnitelmapohjaisuutta ja siihen liittyy tiukka ohjelmistokehitysprosessin kontrollointi. Esimerkiksi NASA:n avaruussukkulaan ohjelmistoja kehittävässä yrityksessä tiukka kehitysprosessin noudattaminen on avain vähäiseen ohjelmiston virhemäärään. Prosessi on tehty siten, että määrittely ja sen muutokset verifioidaan todella tarkasti ennen kuin varsinaiseen ohjelmakoodiin kosketaan. Tämä tarkoittaa suurta määrää erilaisia katselmointeja. Myös muutokset ohjelmakoodissa sekä muutosten syyt on dokumentoitu tarkasti. (Fishman, 2007).

Tämäntyyppinen prosessi on raskas, eikä näin raskaan prosessin pyörittäminen ole taloudellisesti järkevää vähemmän kriittisiä sovelluksia kehitettäessä. Kriittisyyttä mitattiin lähteestä (Taipale et al., 2006) peräisin olevilla kysymyksillä.

## **7 TUTKIMUKSEN TOTEUTUS**

Tutkimus toteutettiin tilastollisena tutkimuksena käyttäen survey-menetelmää. Seuraavissa kappaleissa selitetään menetelmän teoreettista taustaa sekä survey-menetelmään liittyvän survey-lomakkeen reliabiliteetin ja valideetin varmistaminen. Lisäksi selitetään survey-lomakkeen tuottaman datan analyysiin tarvittavat menetelmät.

### ***7.1 Tilastollinen tutkimus***

Tilastollinen tutkimus on systemaattinen tieteellinen tutkimus tilastollisesti mitattavista ominaisuuksista ja ilmiöistä ja niiden suhteista. Kvantitatiivisessa tutkimuksessa korostetaan yleispäteviä syyn ja seurauksen lakeja. Keskeisiä kvantitatiivisessa tutkimuksessa on johtopäätökset aiemmista tutkimuksista, aiemmat teoriat, hypoteesien esittäminen, käsitteiden määrittely, aineiston keruun suunnitelmat, tutkittavien henkilöiden valinta, aineiston saattaminen tilastollisesti käsiteltävään muotoon ja päätelmien teko tilastolliseen analysointiin perustuen. (Hirsjärvi et al., 2004).

### ***7.2 Survey-menetelmä tilastollisen aineiston keräämiseen***

Tutkimusmenetelmäksi valittiin survey-menetelmä, joka on yksi kvantitatiivisen tiedon keruumenetelmä. Survey-menetelmä oli luonnollinen valinta tutkimusmenetelmäksi, koska tutkimuksella haetaan poikkileikkausta testauksen tilasta tutkimuksen organisaatioyksiköissä. Survey on tämän tyyppiseen tutkimukseen perusmenetelmä (Fink ja Kosecoff, 1985).

Survey-menetelmässä tieto kerätään suoraan ihmisiltä. Kerättävä tieto voi olla tunteuksiin, motivaatioon, suunnitelmiin tai uskomuksiin liittyvää. Lisäksi yleensä kerätään henkilön taustatietoa, esimerkiksi ammatista, kokemuksesta ja työn sisällöstä. Yleensä survey toteutetaan kysymyslomakkeen muodossa, jonka henkilö täyttää yk-

sin tai jonkun avustuksella. Lomake voidaan myös täyttää haastatteleamalla puhelimitse tai henkilökohtaisesti. (Fink ja Kosecoff, 1985).

### **7.2.1 Survey-prosessi**

Survey-prosessi sisältää seuraavat vaiheet (Pfleeger ja Kitchenham, 2001):

1. Täsmällisten, mitattavien tavoitteiden asettaminen
2. Survey-prosessin suunnittelu ja aikataulutus
3. Riittävien resurssien saatavuuden varmistaminen
4. Survey-lomakkeen yleisen rakenteen suunnittelu
5. Survey-lomakkeen kysymysten yksityiskohtainen suunnittelu
6. Survey-lomakkeen validointi
7. Osallistujien valinta
8. Kyselyn suorittaminen
9. Surveyn tuottaman datan analysointi
10. Tulosten raportointi

Minun osuuteni tutkimuksesta alkoi kohdasta 6 eli Survey-lomakkeen validointi. Hypoteesit olivat jo muutettu kysymyslomakkeen muotoon. Survey-lomakkeen validoinnilla tarkoitetaan lomakkeen reliabiliteetin ja valideetin varmistamista.

### **7.2.2 Reliabiliteetin varmistaminen**

Luotettavuudella (reliabiliteetillä) tarkoitetaan sitä, että sama mittaus tuottaa aina saman tuloksen. Jos esimerkiksi mittanauha on valmistettu kumista, antaa se mittaus-tulokseksi joka kerta hieman eri tuloksen riippuen siitä kuinka tiukalle mittanauha vedetään. Metallista mittanauhaa käytettäessä tätä ongelmaa ei ole. Tästä huolimatta myös teoriassa täydellisen luotettavaa metallista mittanauhaa käyttämällä voi tulla mittausvirhettä esimerkiksi siten, että mittaaja ei pysty huonon näkönsä takia lukemaan mitan näyttämää tulosta oikein. (Litwin, 1995)

Luotettavuutta voidaan Litwinin (1995) mukaan arvioida seuraavilla viidellä eri tavalla, jotka ovat: 1) testaus-uudelleentestaus (test-retest), 2) vaihtoehtoinen lomake (alternative form), 3) sisäinen yhtenäisyys (internal consistency), 4) haastattelijan luotettavuus (interobserver reliability) sekä 5) haastattelijoiden välinen (intraobserver reliability) luotettavuus.

Testaus-uudelleentestaus (1) tarkoittaa, että samat vastaajat täyttävät saman lomakkeen eri aikoina. Se mittaa mittaustulosten toistettavuutta. Vaihtoehtoinen lomake (2) on sama, mutta siinä kysymykset ja/tai vastaukset kirjoitetaan eri sanoin, jolloin pienennetään niin sanottua oppimisefektiä. Oppimisefekti tarkoittaa, että haastateltavat mahdollisesti muistavat mitä he vastasivat ensimmäisellä kyselykerralla. Tutkimuksessa ei suoraan tämän tyyppistä luotettavuutta varmistettu johtuen siitä, että kyseessä ei ole pitkittäistutkimus. Pitkittäistutkimuksessa samaa lomaketta käytetään useammin kuin kerran ja sillä voidaan mitata esimerkiksi asioiden muutoksia pitkällä aikavälillä. Sen sijaan tutkimuksen suoritus raportoitiin niin tarkasti, että se pystytään uudelleen suorittamaan ja tutkimusryhmän arvion mukaan tulosten pitäisi olla samanlaisia.

Sisäinen yhtenäisyys (3) on indikaattori, joka kertoo kuinka hyvin eri muuttujat mittaavat samaa asiaa. Tämä on tärkeää, koska mikäli yksittäisistä muuttujista koostetaan summamuuttujia, on niiden osoitettava samaan suuntaan, jotta summamuuttuja olisi luotettava. Sisäistä yhtenäisyyttä mitataan Cronbachin alfa-kertoimella. Alfa-kertoimen laskettiin myös tässä tutkimuksessa summamuuttujia tehtäessä.

Haastattelijan luotettavuus (4) mittaa sitä, kuinka samanlaiset tulokset sama haastattelija saa samalta haastateltavalta eri kerroilla. Teoriassa tämän pitäisi olla identtinen. Tätä ei varmistettu johtuen siitä, että tutkimus ei ole pitkittäistutkimus vaan yhden haastattelukerran tulokset riittävät. Mikäli haastattelulomaketta käytettäisiin joskus tulevaisuudessa uudelleen, on epätodennäköistä, että samat henkilöt tekisivät edelleen haastattelut, jolloin tämän tyyppisen luotettavuuden varmistaminen on tarpeellonta.

Haastattelijoiden välinen luotettavuus (5) mittaa kuinka hyvin kaksi tai useampi haastattelijä kysyy saman kysymyksen. Tässä tutkimuksessa tämän varmistaminen ei ole relevanttia, koska haastatteluista yli 80 % suoritti sama henkilö.

### 7.2.3 Validiteetin varmistaminen

Validiteetilla tarkoitetaan sitä, että mittaus mittaa juuri sitä, mitä sen pitäisi mitata. Metallista mittanauhaa käytettäessä tulos on reliaabeli, mutta jos mittanauha on todellisuudessa 2 metriä pitkä vaikka asteikossa lukee 1,9 metriä, niin tulos ei ole validi. (Litwin, 1995).

Validiteettia voidaan Litwinin (1995) mukaan arvioida seuraavilla neljällä eri tavalla, jotka ovat: 1) näennäisvaliditeetti (face validity), 2) sisältövaliditeetti (content validity), 3) kriteerivaliditeetti (criterion validity) ja 4) rakennevaliditeetti (construct validity).

Näennäisvaliditeetti (1) perustuu asiaa tuntemattoman henkilön arvioon surveyn validiteetista. Käytännössä asiaan perehtymättömät ihmiset voivat vain tarkastaa, että kysymykset näyttävät järkeviltä ja että he ymmärtävät mitä niissä kysytään.

Sisältövaliditeetti (2) on näennäisvaliditeettiin verrattuna astetta syvällisempi analyysi, joka perustuu asiantuntijoiden arvioon. Sen avulla voidaan arvioida, sisältääkö survey kaiken tarpeellisen ja ettei se sisällä mitään ylimääräistä. Sisältövaliditeetin arvioimiseksi on myös hyvä käyttää henkilöitä, joita lomakkeella tullaan haastattelemaan, koska siten saadaan pieniin yksityiskohtiin liittyvää palautetta, jota varsinaiset asiantuntijat ei välttämättä ota huomioon.

Kriteerivaliditeetti (3) mittaa, kuinka hyvin surveyn tulokset peilautuvat verrattuna jonkin toisen instrumentin tuottamiin tuloksiin. Kriteerivaliditeetti voidaan jakaa kahteen komponenttiin: samanaikaisvaliditeettiin ja ennustevaliditeettiin. Nimensä mukaisesti samanaikaisvaliditeetti mittaa sitä, kuinka surveyn tulokset peilautuvat jonkun muun vakiintuneen menetelmän samanaikaisesti antamiin tuloksiin. Ennuste-



validiteetti tarkoittaa, kuinka hyvin surveyn tuottama ennuste peilautuu jonkin muun vakiintuneen menetelmän tuottamiin ennustuksiin.

Rakennevaliditeetti (4) on kaikkein syvällisin, mutta samalla kaikkein hankalin arvioida verrattuna edellä mainittuihin validiteetin lajeihin. Se mittaa kuinka tarkoituksen mukainen yksittäinen surveyn muuttuja tai itse survey-lomake on käytännössä, kun sitä käytetään pitkällä aikavälillä. Rakennevaliditeetin ajatellaan koostuvan kahdesta validiteetin muodosta: yhtenevyysvaliditeetista (convergent validity) ja erottelevuusvaliditeetista (divergent validity).

Yhtenevyysvaliditeetilla tarkoitetaan, että eri menetelmät tiedon hankkimiseen tuottavat samat tulokset, eli niiden välinen korrelaatio on korkea. Erottelevuusvaliditeetti taas tarkoittaa sitä, että survey-lomakkeella mitatut asiat erottuvat läheisistä, mutta kuitenkin eri käsitteistä tai asioista, eli niiden välinen korrelaatio on matala.

Sisältövaliditeetti varmistettiin tässä tutkimuksessa yhteistilaisuudessa TKK:n tutkijoiden (TKK: Mika Mäntylä, Juha Itkonen, Jari Vanhanen ja Antti Hätinen, LTY: Jussi Kasurinen, Ossi Taipale, Kari Smolander ja Riku Luomansuu) kanssa, jotka antoivat täsmennysehdotuksia kysymyksiin sekä ehdottivat muutaman kysymyksen lisäystä, jotka lomakkeeseen lopulta lisättiin. Lisäksi lomakkeelle tehtiin pilottitesti, joka tarkoittaa lomakkeen testaamista todellisessa suoritussympäristössä (Litwin, 1995). Pilottitesti toteutettiin erään tutkimusryhmän jäsenen ja yhden tutkimukseen osallistuneen organisaatioyksikön kanssa. Tämä auttoi varmistamaan sen, että haastatteluun varattu aika todella riittää kysymysten läpikäymiseen ja että haastateltavat ymmärtävät, mitä kysytään. Pilottitestausta siis auttoi näennäisvaliditeetin sekä osittain myös sisältövaliditeetin varmistamisessa.

Kriteerivaliditeetti varmistetaan tutkimuksessa vertaamalla tämän tutkimuksen tuloksia samanaikaisesti toteutettavan laadullisen tutkimuksen tuloksiin niin sanotulla triangulaatiolla. Menetelmä triangulaatiolla tarkistetaan, antavatko eri menetelmät toistensa tuloksia vahvistavia tai heikentäviä tuloksia. Sen käsittely on rajattu tämän diplomityön aihealueen ulkopuolelle.

Rakennevaliditeetin mittaaminen ei tämän tutkimuksen aikana onnistu, koska käytetty survey-lomake on uusi ja kehitetty tätä tutkimusta varten. Mikäli samaa lomaketta käytetään useissa tutkimuksissa peräkkäin, sen varmentaminen tulevaisuudessa on mahdollista.

### **7.3 Tulosten analyysiin tarvittavat menetelmät**

Kysymyslomakkeessa käytetyt kysymykset voidaan luokitella avoimiin ja strukturoituihin kysymyksiin (Fink ja Kosecoff, 1985). Strukturoiduissa kysymyksissä vastaukset on ennalta määriteltäviä, joista vastaaja valitsee sopivimman. Avoimissa kysymyksissä vastaaja muodostaa vastauksen itse.

Strukturoituna kysymyksiä tutkimuksessa käytettiin 5 portaisia Likert-muuttujia, joista vastaus 1 merkitsi ”täysin eri mieltä”, 5 ”täysin samaa mieltä” ja muut vaihtoehdot olivat tältä väliltä. Lisäksi strukturoituna kysymyksiä käytettiin monivalintakysymyksiä, joista vastaaja valitsi yhden tai useamman vastauksen. Tutkimuksessa Likert-muuttujista tehtiin myös uusia summamuuttujia.

Likert-muuttujat voidaan mieltää joko intervalli- tai ordinaali-tyyppisiksi (Fink ja Kosecoff, 1985). Tässä tutkimuksessa ne on ajateltu intervalli-tyyppisiksi eli numeroitten etäisyydellä on merkitystä ja ne ovat säännöllisen välimatkan päässä toisistaan, eli ero vastauksen 1 ja 2 välillä on sama kuin ero vastausten 4 ja 5 välillä. Samoin ero vastausten 1 ja 3 välillä on sama kuin ero vastausten 2 ja 4 välillä. Myös tutkimuksessa käytetyt monivalintakysymykset ovat intervalli-tyyppisiä. Muuttujien tyyppillä on merkitystä siksi, että ordinaali- ja intervalliasteikon muuttujia käsitellään erilaisilla tilastollisilla menetelmillä.

Seuraavissa kappaleissa on kuvattu kysymyslomakkeen tietojen analysointiin tarvittavat matemaattiset menetelmät sekä avoimien kysymysten luokitteluun tarvittava menetelmä.

### 7.3.1 Parittainen t-testi

Parittaista t-testiä (paired t-test) käytetään vertailemaan kahden eri otoksen keskiarvoa. Tutkimuksessa tarvittavana t-testi on parittainen, koska molemmat keskiarvot on laskettu samasta otannasta. T-testillä testataan, onko keskiarvoilla tilastollisesti merkittävää eroa, vai voiko ero olla pelkästään sattumasta kiinni. Parittaiseen t-testiin liittyy oleellisesti riskitaso, joista yleisesti käytettyjä riskitasoja ovat 0,05, 0,01 ja 0,001.

Parittaisessa t-testissä testataan nollahypoteesia vastakkaiseen hypoteesiin. Nollahypoteesissa  $H_0$ : Keskiarvojen välillä ei ole eroa ja vastakkaisessa hypoteesissa  $H_1$ : Keskiarvojen välillä on eroa.

Mikäli testin merkitsevyydeksi saadaan pienempi kuin käytetty merkitsevyydestaso (esimerkiksi 0,05), nollahypoteesi  $H_0$  hylätään eli keskiarvot poikkeavat tilastollisesti toisistaan käytetyllä riskitasolla, (Frank, 2004).

### 7.3.2 Cronbachin alfakerroin

Jotta uudet summamuuttujat kuvaisivat hyvin mitattavaa asiaa, on niiden taustalla olevien yksittäisten muuttujien osoitettava samaan suuntaan. Tämän varmistukseksi laskettiin Cronbachin alfakerroin.

”Cronbachin alfa lasketaan muuttujien välisten keskimääräisten korrelaatioiden ja väittämien lukumäärän perusteella. Mitä suurempi alfan arvo on, sitä yhtenäisempi mittarin voidaan katsoa olevan.” (Saaranen et al., 2008). Hyvä Cronbachin alfakerroin riippuu siitä, kuinka mittauksia käytetään. Dybån mukaan kerroin arvojen 0.7 ja 0.8 välillä on riittävä. (Dybå, 2000)

### 7.3.3 Kolmogorov-Smirnov-testi

Kolmogorov-Smirnov-testi testaa, onko aineisto normaalijakautunut vertaamalla aineistoa teoreettiseen määriteltyyn jakaumaan, joka voi olla esimerkiksi normaalijakautunut tai Poisson-jakautunut. Testaus täytyy tehdä siksi, koska jäljempänä tehtävä Pearsonin korrelaatiokertoimen laskeminen edellyttää laskennan muuttujilta normaalijakaumaa. Jotta muuttujan normaalijakautuneisuus voitaisiin osoittaa, Kolmogorov-Smirnov testissä yritetään merkitsevyydeksi saada mahdollisimman iso luku, jotta ei voitaisi kumota nolla-hypoteesia, joka on, että muuttujan jakauman ja normaalijakauman välillä ei ole eroa. Myös Kolmogorov-Smirnov testiin liittyy oleellisesti riskitaso, joista yleisesti käytettyjä riskitasoja ovat 0,05, 0,01 ja 0,001.

Kolmogorov-Smirnov testissä testataan nollahypoteesia vastakkaiseen hypoteesiin. Nollahypoteesissa  $H_0$ : Muuttujan jakauman ja normaalijakauman välillä ei ole eroa ja vastakkaisessa hypoteesissa  $H_1$ : Muuttujan jakauma ja normaalijakauma poikkeavat.

Mikäli esimerkiksi 0,05 riskitasoa käytettäessä testin merkitsevyydeksi saadaan suurempi kuin 0,05, jää nollahypoteesi  $H_0$  voimaan eli aineisto voidaan olettaa normaalijakautuneeksi 0,05 ja sitä pienemmillä riskitasoilla (Frank, 2004).

### 7.3.4 Pearsonin korrelaatiokerroin ja korrelaatiokertoimen neliö

Pearsonin korrelaatiokerroin on vähintään kahden intervalliasteikollisen muuttujan keskinäisen lineaarisen riippuvuuden voimakkuutta kuvaava tilastollinen tunnusluku, jonka arvo vaihtelee välillä -1... +1. Korrelaatiokertoimen ollessa 0, ei muuttujien välillä ole lineaarista riippuvuutta. Vastaavasti arvoilla (+/-) 1 muuttujien välillä on täydellinen positiivinen / negatiivinen lineaarinen riippuvuus. Toinen tärkeä tunnusluku on Pearsonin korrelaatiokertoimen neliö  $R^2$ . Esimerkiksi jos  $R^2 = 0,32$  sanotaan, että selittävä muuttuja selittää 32 % selitettävän muuttujan varianssista (Sivonen, 2004).

### 7.3.5 Lineaarinen regressioanalyysi

Lineaarinen regressioanalyysi on tilastollinen analyysimenetelmä, jossa aineiston perusteella arvioidaan selittävien muuttujien vaikutusta selitettävään muuttujaan. Menetelmällä pyritään löytämään suoran yhtälö (regressiosuora), joka kuvaa riippuvuutta mahdollisimman hyvin. Yleisesti tämä toteutetaan niin sanotun pienimmän neliösumman menetelmällä.

Lineaarisen regressioanalyysin tuloksena on suoran yhtälö, joka on muotoa:  $Y = a + b_1X_1 + b_2X_2$ , missä  $Y$  on selitettävän muuttujan arvo,  $a$  vakiotekijä,  $X_1$  ja  $X_2$  selittävät muuttujat sekä  $b_1$  ja  $b_2$  niiden regressiokertoimet. Esimerkissä selittäviä muuttujia on 2, mutta niitä voi olla kuinka paljon tahansa (Mattila, 2008).

### 7.3.6 LB (Like Best) tekniikka avointen kysymysten luokitteluun

Survey-lomakkeessa olevien avoimien kysymysten luokitteluun käytettiin Finkin ja Kocehoffin mukaista LB (Like Best) tekniikkaa, (Fink ja Kosecoff, 1985). Tekniikassa vastaajia pyydetään listaamaan asioita, joiden määrälle asetetaan ala- ja yläraja. Vastauksia analysoitaessa ne luokitellaan sopiviin luokkiin. Finkin mukaan on mahdollista tehdä sopivat luokat vastausten perusteella tai tehdä luokat etukäteen. Mielestäni tutkimuksen kannalta uskottavampi tapa on luoda luokat kirjallisuuden perusteella etukäteen, koska tällöin luokat tulee samalla sidottua teoriaan ja näin myös tutkimuksessa toimittiin.

## 8 TUTKIMUKSEN TAUSTATIEDOT

Seuraavissa kappaleissa esitetään tutkimuksen taustatiedot, jotka koostuvat populaatiosta ja otoksesta, tutkimuksen rajauksesta, tutkimusdatan keräämistekniikasta, haastatteluajasta, haastateltujen henkilöjen jakaumasta vastuualueen mukaan, haastateltujen organisaatioyksiköiden jakaumasta toimialan mukaan sekä tietoja liittyen henkilöistöön, käytettyihin menetelmiin ja testauksen automaatioon.

### **8.1 Tutkimuksen populaatio ja otos**

Tämän tutkimuksen populaatio koostui organisaatioyksiköistä, jotka Suomessa kehittävät ja testaavat keskikriittisiä sovelluksia. Populaatio näin ollen rajaa ulkopuolelle erittäin kriittisiä sovelluksia tekevät yritykset (esimerkiksi ydinvoimaloihin tehtävät sovellukset) sekä todella yksinkertaisia sovelluksia tekevät yritykset. Määritelmän mukainen populaatio ei ole kooltaan kovin suuri, mutta täsmällistä numeroarvoa populaation kooksi ei voida antaa.

Tässä tutkimuksessa käytetty organisaatioyksikkö (OU) on määritelty standardissa ISO/IEC 15504-1 ja sen mukaan organisaatioyksikkö käyttää yhtä tai useampaa prosessia, joilla on yhtenevät liiketoiminnalliset päämäärät, (ISO/IEC, 2002). Prosessi tarkoittaa standardin mukaan sarjaa toisiinsa liittyviä aktiviteetteja, jotka muuttavat panokset tuotoksiksi. Organisaatioyksikkö on tyypillisesti suuren organisaation osa, mutta se voi olla myös pieni organisaatio kokonaisuudessaan.

Finkin mukaan otos on osa populaatiosta (Fink, 2003). Hyvä otos on ominaisuuksiltaan samanlainen kuin populaatio minkä osa se on, se on ainoastaan pienempi. Finkin mukaan satunnaisotanta (probability sampling) tuottaa tämäntyyppisen populaation osan. Yksi satunnaisotannan muoto on systemaattinen otanta, missä esimerkiksi joka toinen tai kolmas alkio poimitaan populaatiosta.

Tutkimuksen otos luotiin valitsemalla satunnaisotannalla populaation tietokannasta (käytännössä Excel lomakkeesta) joka toinen yritys. Populaation tietokanta on muo-

dostettu yhteistyössä TEKES:in ja TE-keskuksen kanssa ja se sisältää populaation määritelmän mukaiset yritykset. Haastatteluun valittiin vain yksi organisaatioyksikkö organisaatiota kohti, jotta vältettäisiin suurien yritysten aiheuttama vääristymä otoksessa.

Tutkimukseen osallistumispyyntö lähetettiin sähköpostitse valittuihin 30 organisaatioyksikköön. Sähköpostin lähettämisen jälkeen jokaiseen organisaatioyksikköön soitettiin ja kysyttiin halua osallistua tutkimukseen. Suoria vastauksia halusta tutkimukseen osallistumiseen sähköpostilla tuli 7 kpl. Puhelimitse saimme 11 osallistujaa. Lisäksi tutkimuksen ensimmäiseen vaiheeseen osallistuneet 12 organisaatioyksikköä osallistuivat automaattisesti toiselle haastattelukierrokselle, joten kokonaisuudessaan tutkimukseen osallistuneille yrityksille tuli 31. Vastausprosentiksi saatiin  $31/42 * 100 \% = 74 \%$

Tutkimusryhmän arvion mukaan otoskoko riittää tulosten perusteella tehtävään päätelyyn ja yleistämiseen populaation määritelmän mukaisiin yrityksiin. On hyvä myös huomata, että otoskoon kasvattamisella ei voida kasvattaa tutkimustulosten yleistettävyyden tarkkuutta (Baskerville ja Lee, 2003). Mikäli otoskoko kasvatettaisiin esimerkiksi 3-kertaiseksi, ei tuloksia voisi yleistää sen suuremmalla tarkkuustasolla. Ainoastaan koko populaation läpikäyminen riittäisi tulosten tarkkaan yleistettävyyteen.

## **8.2 Tutkimuksen rajaus**

Tutkimuksen populaatio ja otos ovat rajattuja ja tulokset ovat yleistettävissä vain vastaaviin organisaatioyksiköihin.

## **8.3 Tutkimusdatan keräämistekniikka**

Tutkimuksessa päädyttiin haastattelemaan osallistuvien organisaatioyksiköiden edustajia henkilökohtaisesti. Vaihtoehtoina olisi ollut kyselylomakkeen lähettäminen sähkö-

köpostitse tai puhelinhaastattelu. Molemmat vaihtoehdot ovat henkilökohtaiseen haastatteluun nähden kustannustehokkaampia vaihtoehtoja, mutta niissä on tiettyjä ongelmia.

Kyselylomakkeen lähettämisessä on ongelmana, että vastausten lukumäärä (response rate) jää usein alhaiseksi, joka voi aiheuttaa tutkimukseen vääristymiä ja tätä kautta heikentää tutkimuksen uskottavuutta. Hirsjärven mukaan kohdentamattomalle joukolle lähetettynä parhaimmillaankin vastauksia saadaan 30–40 prosenttia lähetetyistä lomakkeista. Kohdennetulle ryhmälle vastausprosentti voi nousta jopa 70–80 prosenttiin, mikäli lomakkeita ”karhutaan” muutamaan otteeseen (Hirsjärvi et al., 2004). Periaatteessa korkea vastausten lukumäärä olisi tässä tapauksessa ollut mahdollista, koska ryhmä oli kohdennettu.

Kyselylomaketta vastaanottaessa ei kuitenkaan pystytä verifioimaan, kuka lomakkeen vastauksen on täyttänyt. On mahdollista, että lomakkeen vastaanottanut henkilö delegoi lomakkeen täyttämisen esimerkiksi kesätyöntekijälle. Kolmantena ongelmana on, että vastaajat jättävät täyttämättä osan kohdista, jolloin tilastollisten tunnuslaskujen laskeminen ei välttämättä ole mahdollista. Neljäntenä ongelmana on, että vastaajat ymmärtävät kysymykset eri tavalla. Henkilökohtaisesti haastatteleamalla pystytään kysymyksiin antamaan lisätietoa ja vastaamaan haastateltavan mahdollisesti esittämiin selventäviin kysymyksiin. Tutkimuksessa käytetty kysymyslomakepohja oli englanniksi, koska yksi lomakkeen laatijoista oli intialainen. Tämän johdosta mahdollisuus monitulkintaisuuteen kasvoi entisestään.

Puhelinhaastattelu on kyselylomakkeen lähettämiseen nähden astetta parempi vaihtoehto. Sillä ratkaistaan kaikki edellä mainitut kyselylomakkeen lähettämiseen ja vastaanottamiseen liittyvät ongelmat. Ongelmana tässä on kuitenkin interaktion rajallisuus. Puhelimen kautta haastateltaessa ei näe vastaajan elekieltä, josta haastattelija pystyy välillä päättelemään, miten haastateltava kysymyksen ymmärsi. Surveylomake oli myös varsin pitkä koostuen yli sadasta erillisestä vastattavasta kohdasta. Puhelimitse haastattelu ei muutenkaan ole yhtä sujuvaa kuin henkilökohtainen haastattelu johtuen muun muassa äänen laadusta. Haastattelun nauhoitus on edelleen vai-



keampaa samasta syytä johtuen. Myös verkostoja on helpompi luoda henkilökohtaisesti haastattelemalla. Tällä on merkitystä tulevaisuudessa toteutettavia tutkimuksia silmälläpitäen.

Haastattelun huonoina puolina on, että siihen katsotaan sisältyvän monia virhelähteitä johtuen haastattelijasta, haastateltavasta ja itse haastattelutilanteesta. Haastattelijasta johtuvia virhelähteitä on käsitelty jo aikaisemmin tässä työssä kappaleessa: ”7.2.2 Reliabiliteetin varmistaminen”. Haastateltavasta johtuva virhelähde on, että haastattelu on tilannesidonnaista, jolloin haastateltava jossain toisessa tilanteessa saattaisi antaa erilaisia vastauksia. Haastattelutilanteesta johtuva virhelähde on esimerkiksi se, että haastattelussa on taipumus antaa sosiaalisesti suotavia vastauksia. (Hirsjärvi et al., 2004). Tässä tapauksessa se voisi merkitä yrityksen käytäntöjen liioittelua. Tutkimuksen otanta on kuitenkin sen verran suuri, että mainituista virhelähteistä johtuvat virheet pitäisi hävitä tilastollisesti merkitsevän, riittävän suuren otannan myötä.

#### **8.4 Haastattelu aika**

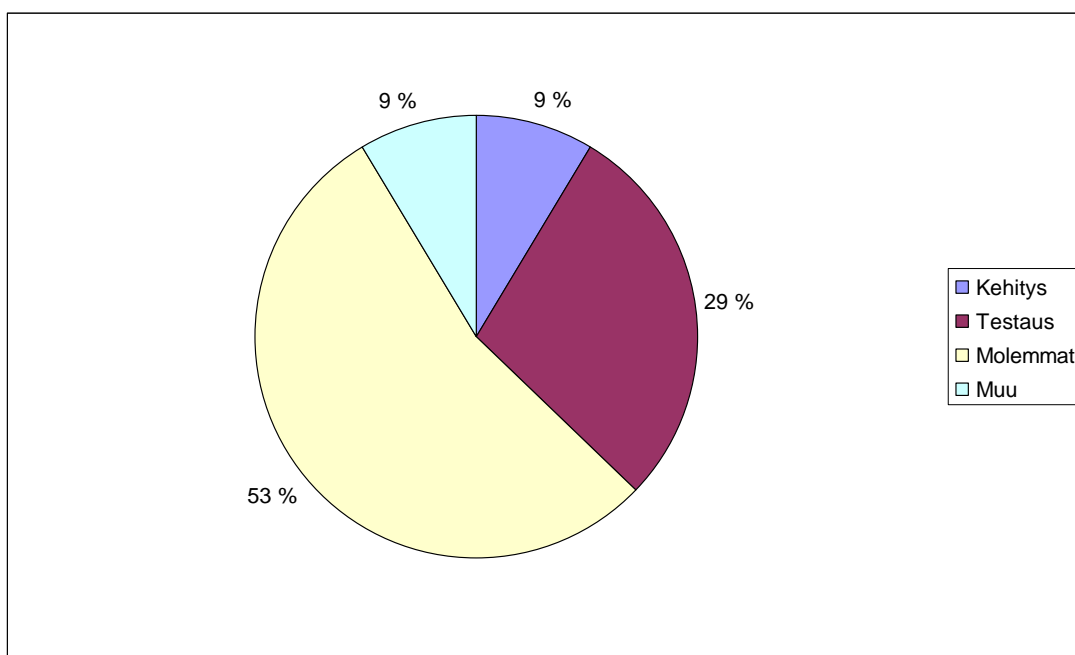
Tutkimusaineisto kerättiin 5.12.2008 – 3.2.2009 välisenä aikana haastattelemalla organisaatioyksikköjen edustajia. Survey-lomake on liitteenä 1. Haastattelu myös nauhoitettiin mahdollista myöhempää tarkastelua varten, mikäli haastatteliija antoi siihen luvan. Yksi haastatelluista henkilöistä kielsi haastattelun nauhoituksen.

Haastatteluun pyydettiin varaamaan aikaa yksi tunti. Toteutunut keskiarvo haastatteluajaksi oli yksi tunti ja kolme minuuttia. Lyhimmillään haastattelu vei 28 minuuttia ja pisimmillään kaksi tuntia. Haastattelujen venymistä aiheutti joidenkin kysymysten monitulkintaisuus, joiden selventäminen vei aikaa.

## 8.5 Haastatellut henkilöt

Haastateltaviksi haimme kehityksestä, testauksesta tai molemmista vastaavaa henkilöitä. Kuvassa 13 on esitetty jakauma haastateltavan vastualueen mukaan. Haastateltavien vastualueet jakautuivat siten, että sekä kehityksestä että testauksesta vastasi yli puolet eli 53 % haastatelluista. Tätä voidaan pitää hyvänä asiana, sillä vastauslomakkeen kysymykset edellyttivät koko ohjelmistokehitysprosessin tuntemista, ja todennäköisemmin molemmista alueista vastuussa oleva henkilö tuntee sen parhaiten.

Testauksesta vastasi 29 % ja kehityksestä vastasi 9 % haastatelluista. Kategoriaan ”Muu” (9 %) kuuluivat seuraavat vastualueet: vastuu kehitysympäristöstä, vastuu tuotteesta sekä vastuu ohjelmistojen myynnistä.

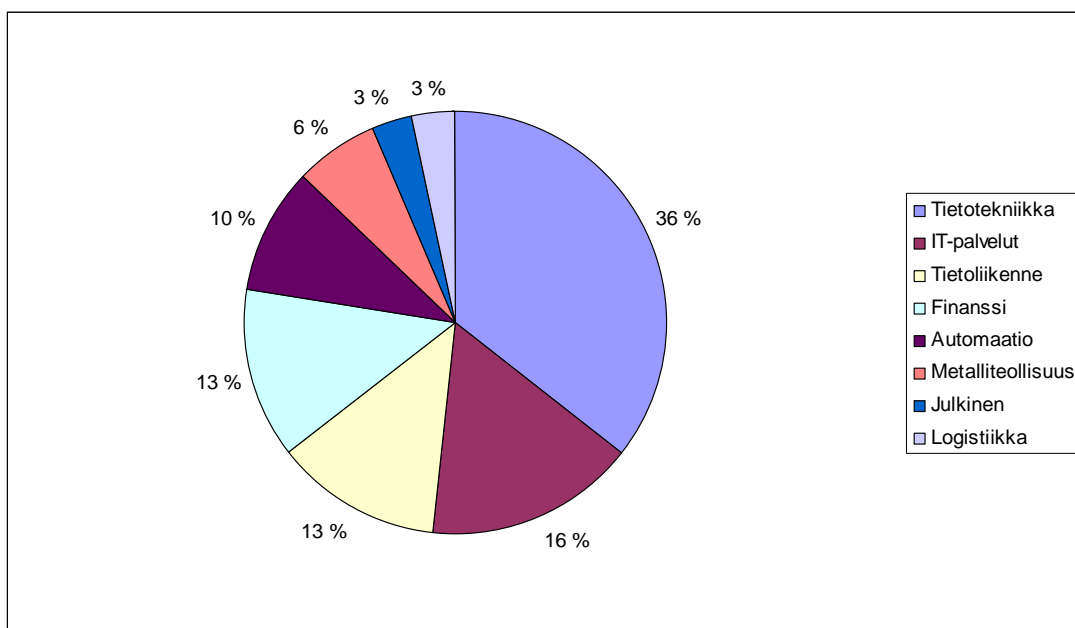


Kuva 13. Jakauma haastateltavan vastualueen mukaan.

## 8.6 Haastatellut organisaatioyksiköt

Kuvassa 14 on esitetty jakauma haastateltujen organisaatioyksiköiden toimialan mukaan. Haastatellut organisaatioyksiköt jakautuivat toimialoittain siten, että IT-alan yrityksen muodostivat haastatelluista suurimman osan (65 %). IT-alalla toimivat organisaatioyksiköt on jaettu Tietotekniikka (36 %), Tietoliikenne (13 %) ja IT-palvelut toimialoihin (16 %) siten, että Tietotekniikka-toimialaan kuuluvat organisaatioyksiköt, joilla on oma tuote, Tietoliikenne-toimialaan kuuluvat organisaatioyksiköt joiden tuote tai tarjottavan palvelun ydinosaminen liittyy tietoliikenteeseen ja IT-palvelut toimialaan organisaatioyksiköt, jotka tuottavat tietoteknisiä palveluita, mutta jotka eivät kuitenkaan kuulu kahteen edellä mainittuun toimialaan.

Muita haastatteluissa edustettuja toimialoja olivat: Finanssi (13 %), Automaatio (10 %), Metalliteollisuus (6 %), Julkinen (3 %) ja Logistiikka (3 %).



Kuva 14. Jakauma toimialan mukaan.

Haastateltujen yritysten henkilöstöön, käytettyihin menetelmiin ja testauksen automaatioon liittyvien tietojen maksimi, minimi ja mediaani ovat esitettynä taulukossa 5. Mediaania käytetään keskiarvon sijaan, koska yritysten suuresta kokoerosta johtu-

en suurimman tutkimukseen osallistuneen yrityksen arvot olisivat vääristäneet tuloksia.

**Taulukko 5. Tietoa henkilöstöstä, käytetyistä menetelmistä ja testausautomaatiosta**

	<b>Maksimi</b>	<b>Minimi</b>	<b>Mediaani</b>
Henkilöstön lukumäärä	350000	4	315
Kehittäjien ja testaajien lukumäärä organisaatioyksikössä	600	0	30
Testausautomaation osuus %	90	0	10
Ketterien menetelmien osuus käytetyistä menetelmistä %	100	0	30
Testaajien lukumäärä suhteessa tarpeeseen %	100	10	75
Kuinka suuri osa kehitysjajasta käytetään testaukseen %	70	0	25

## 9 HYPOTEESIEN TESTAUKSEN TULOKSET

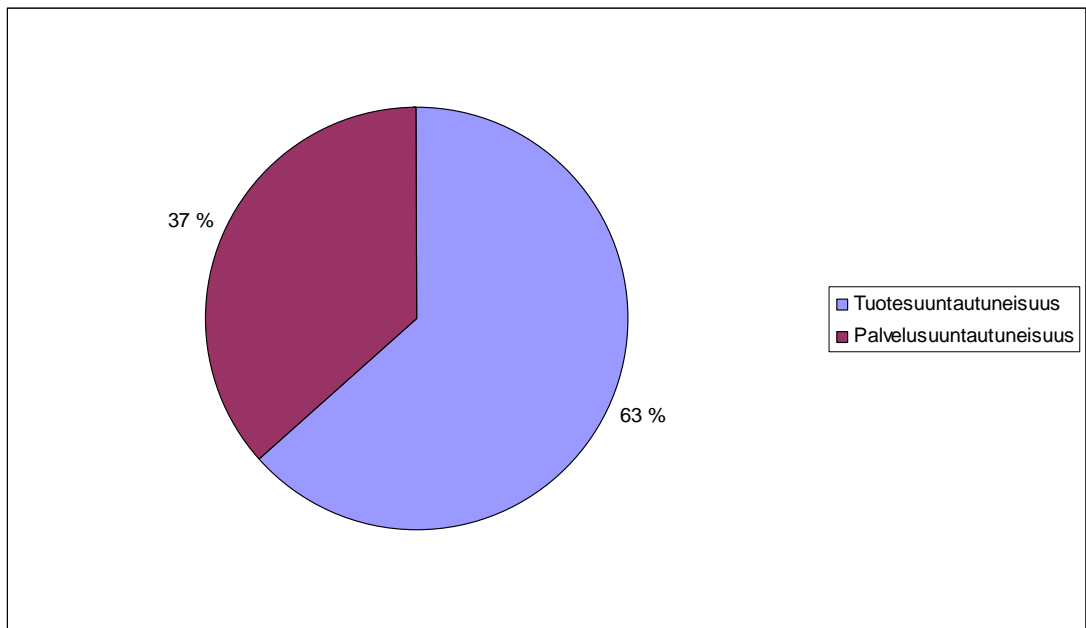
Tutkimuksen hypoteesit testattiin SPSS-ohjelmaa käyttäen, (SPSS, 2004), joka on tarkoitettu tilastolliseen analysointiin. Analysointi jakautui neljään vaiheeseen jotka ovat:

- 1) Mallin muuttujien rakentaminen ja niiden sisäisen eheyden laskeminen
- 2) Muuttujien normaalijakautuneisuuden selvittäminen
- 3) Korrelaation laskeminen
- 4) Lineaarisen regressioanalyysin suorittaminen

Seuraavissa kappaleissa esitetään hypoteeseihin liittyvien osakysymysten jakaumat ja keskiarvot sekä yllämainitun vaiheistuksen mukaisesti rakennetaan muuttujat, selvitetään niiden normaalijakautuneisuus sekä lasketaan korrelaatiomatriisi ja suoritetaan regressioanalyysi.

### **9.1 Liiketoimintasuuntautuneisuus**

Liiketoimintasuuntautuneisuuteen liittyvät kysymykset ovat lomakkeen kysymykset 5.1 – 5.9. Kysymykset 5.1–5.4 liittyvät tuotteen tuottamiseen ja kysymykset 5.5–5.9 palvelun tuottamiseen. Organisaatioyksiköiden liikevaihto jakautui siten, että 63 % organisaatioyksiköistä on tuotesuuntautuneita ja 37 % palvelusuuntautuneita, kuva 15.



**Kuva 15. Jakauma liiketoimintasuuntautuneisuuden mukaan**

Tuotteen ja palvelun tuottamisen määrän suhdelvusta muodostettiin liiketoimintasuuntautuneisuutta kuvaavat muuttujat.

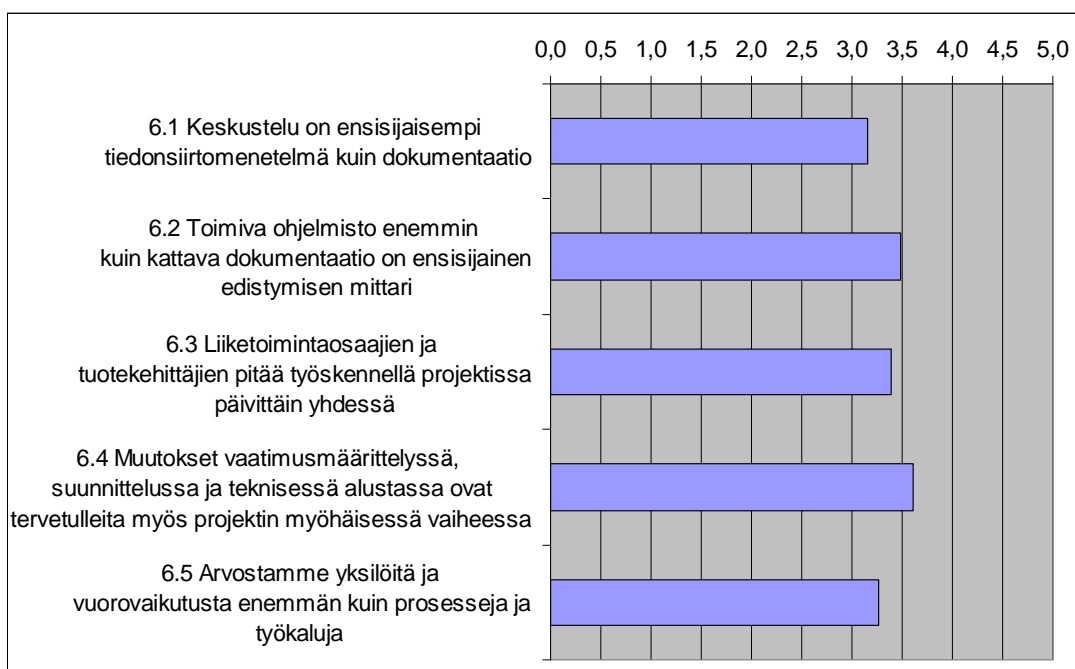
Muodostettujen muuttujien kaavat ovat:

$$\text{Palvelusuuntautuneisuus-\%} = (5.5+5.6+5.7+5.8 / 5.1+5.2+5.3+5.4+5.5+5.6+5.7+5.8) * 100$$

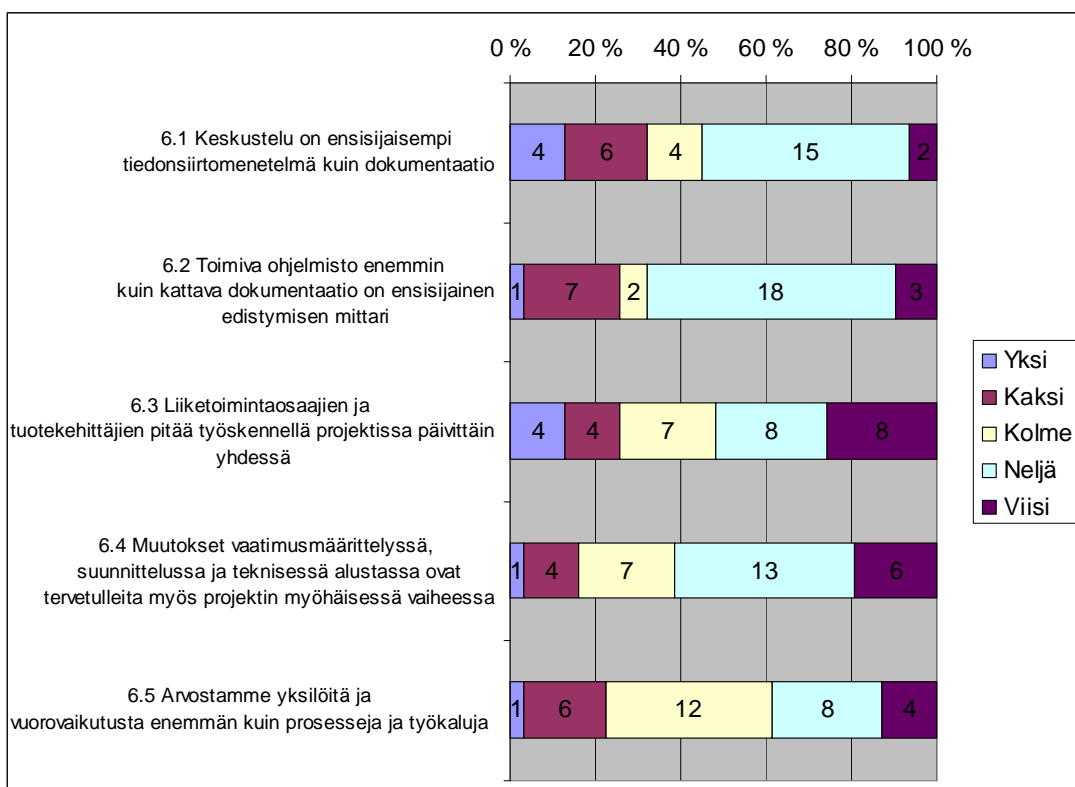
$$\text{Tuotesuuntautuneisuus-\%} = (5.1+5.2+5.3+5.4 / 5.1+5.2+5.3+5.4+5.5+5.6+5.7+5.8) * 100$$

## **9.2 Ohjelmistokehitysmenetelmä**

Ohjelmistokehitysmenetelmään liittyvät kysymykset ovat lomakkeen kysymykset 6.1 - 6.5. Haasteltavia pyydettiin arvioimaan käytettyä ohjelmistokehitysmenetelmää 5-portaisella Likert-asteikolla vastaamalla ohjelmistokehitysmenetelmiin liittyviin väittämiin. Vastaus 1 merkitsi täysin eri mieltä ja 5 täysin samaa mieltä. Kuvassa 16 on esitetty osakysymysten keskiarvo ja kuvassa 17 osakysymysten jakauma.



Kuva 16. Ohjelmistokehitysmenetelmiin liittyvien osakysymysten vastausten keskiarvot



Kuva 17. Ohjelmistokehitysmenetelmiin liittyvien osakysymysten vastausten jakaumat

Teoriassa vastaus 5 viittaa ketterään ohjelmistokehitysmenetelmään ja vastaus 1 suunnitelmapohjaiseen menetelmään. Vastajaat tuntuivat painottavan ketteryyttä,

koska kaikkien osakysymysten vastaukset olivat yli 3 ja myös jakauma painottuu vastauksilla 4 ja 5.

Tuloksia analysoidessa huomasimme, että harvalla tutkimukseen osallistuneista organisaatioyksiköistä kaikki kysymykset olisivat osoittaneet samaan suuntaan. Tämä johtuu siitä, että haastatelluista organisaatioyksiköistä lähes kaikki käyttivät ohjelmistokehitysmenetelmänä niin sanottua hybridimallia, jossa on piirteitä sekä suunnitelmepohjaisista että ketteristä menetelmistä. Näin ollen osakysymykset eivät voi osoittaa samaan suuntaan.

Cronbachin alfa kertoimen laskemalla havaitsimme, että kysymykset 6.1 ja 6.4 ovat samansuuntaisia Cronbachin alfa kertoimen ollessa pyöristettynä 0.7 eli sisäinen eheys on riittävällä tasolla, taulukko 6. Näistä kahdesta kysymyksestä näin ollen muodostettiin muuttuja, joka kuvaa käytettyä ohjelmistokehitysmenetelmää.

Muodostetun muuttujan kaava on:

$$\text{Ohjelmistokehitysmenetelmä} = (6.1+6.4)/2$$

**Taulukko 6. Cronbachin alfa kerroin kysymyksille 6.1–6.5**

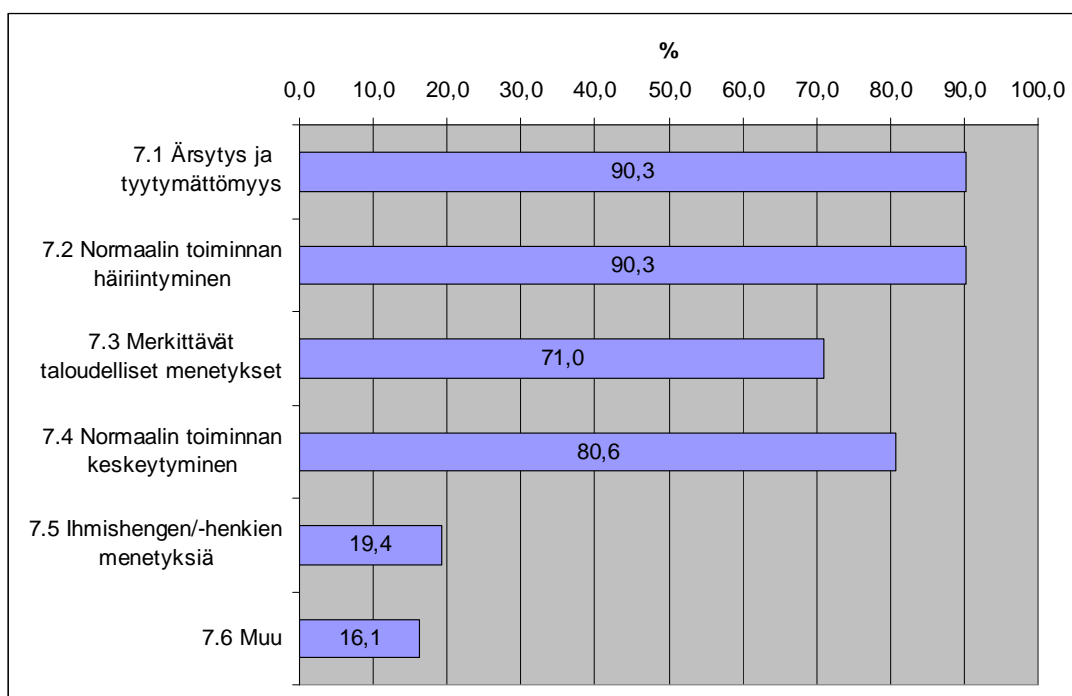
Ohjelmistokehitysmenetelmä		
Kysymykset	Muuttujien lukumäärä	Kysymysten välinen Cronbachin alfa
6.1, 6.2, 6.3, 6.4 ja 6.5	5	0.554
6.1, 6.2, 6.3 ja 6.4	4	0.565
6.1, 6.2 ja 6.4	3	0.630
6.1 ja 6.4	2	0.691

### **9.3 Kriittisyys**

Yrityksen lopputuotteen kriittisyys oli monivalintakysymys, johon liittyvät kysymykset ovat lomakkeen kysymykset 7.1 – 7.6. Vastaaajia pyydettiin arvioimaan heidän tuotteidensa vioista aiheutuvat ongelmat ja rastittamaan lomakkeen monivalinta-



kysymyksessä kaikki soveltuvat kohdat. Tutkimuksessa tarkastellaan tuotteiden vi-  
oista välittömästi aiheutuvia ongelmia. Kuvassa 18 on esitetty vastausten jakauma,  
jossa virheen vakavuus kasvaa pystyakselia pitkin alaspäin mentäessä. 90 % vastaa-  
jista arvioi, että vika heidän tuotteessaan aiheuttaa ärsytystä ja tyytymättömyyttä sekä  
yksilön tai yrityksen normaalin toiminnan häiriintymistä. 70 % arvioi, että tuotteen  
vian aiheuttavan merkittäviä taloudellisia menetyksiä. Yksilön tai yrityksen normaali-  
nin toiminnan häiriintymisen arvioi 80 % vastaajista ja ihmishenkien menetyksen  
19,4 % vastaajista. Kategoriaan ”muu” (16,1 %) kuuluvat seuraavat ongelmat: yh-  
teiskunnan toiminnan vaarantuminen, armeijan, lentoliikenteen tms. taajuudet voivat  
häiriintyä, maineen menetys ja huono näkyvyys mediassa.



**Kuva 18. Vikojen aiheuttamat ongelmat**

Kysymyksistä muodostettiin kriittisyyttä kuvaava summamuuttuja siten, että suu-  
rempaa aiheutuvaa ongelmaa painotettiin suuremmalla kertoimella, esimerkiksi ih-  
mishenkien menetystä painotettiin kertoimella 4 ja ärsytystä ja tyytymättömyyttä ker-  
toimella 1. Muuttuja lisäksi skaalattiin asteikolle 0-5 jakamalla se kolmella.

Muodostetun muuttujan kaava on:

$$\text{Kriittisyys} = (7.1 + 2*7.2 + 3*7.3 + 4*7.4 + 5*7.5)/3$$

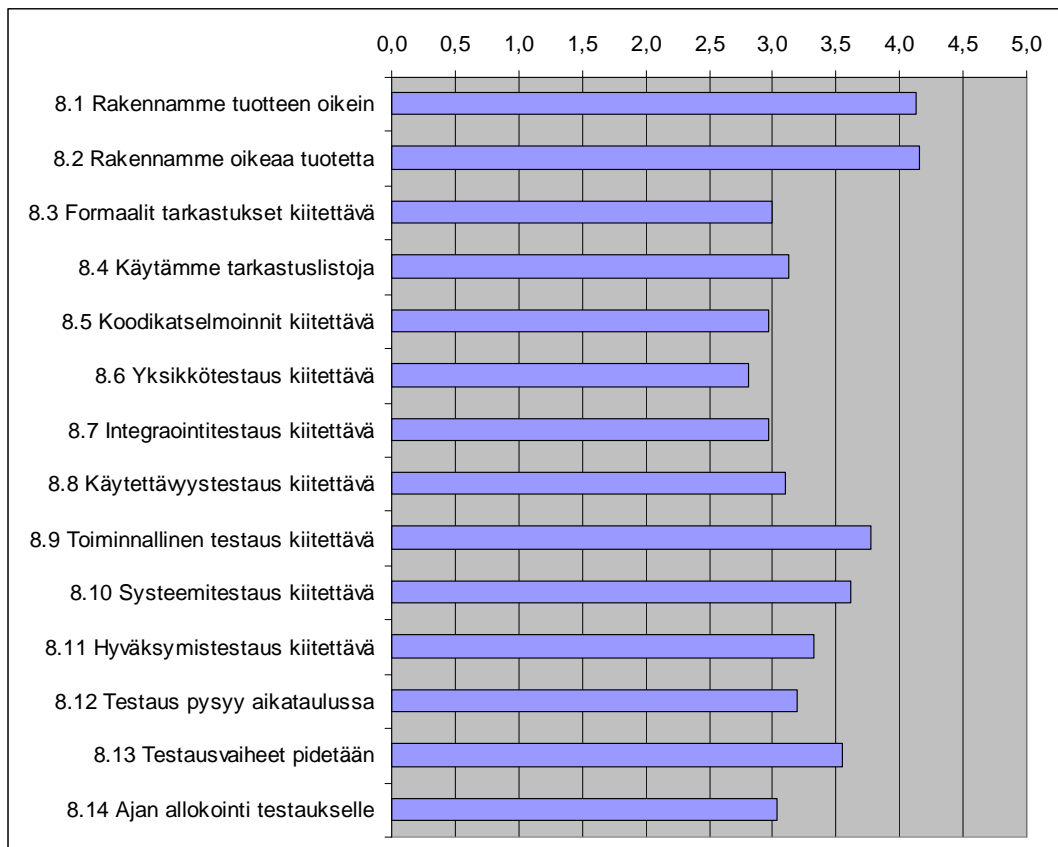
## **9.4 Ohjelmistotestauksen taso**

Testauksen tasoon liittyvät kysymykset ovat lomakkeen kysymykset 8.1 – 8.11. Haasteltavia pyydettiin arvioimaan ohjelmistotestauksen tasoa 5-portaisella Likert-asteikolla vastaamalla testaukseen liittyviin väittämiin. Vastaus 1 merkitsi täysin eri mieltä ja 5 täysin samaa mieltä. Ohjelmistotestausta mittaavien kysymysten keskiarvot ja jakaumat ovat kuvien 19 ja 20 mukaisia.

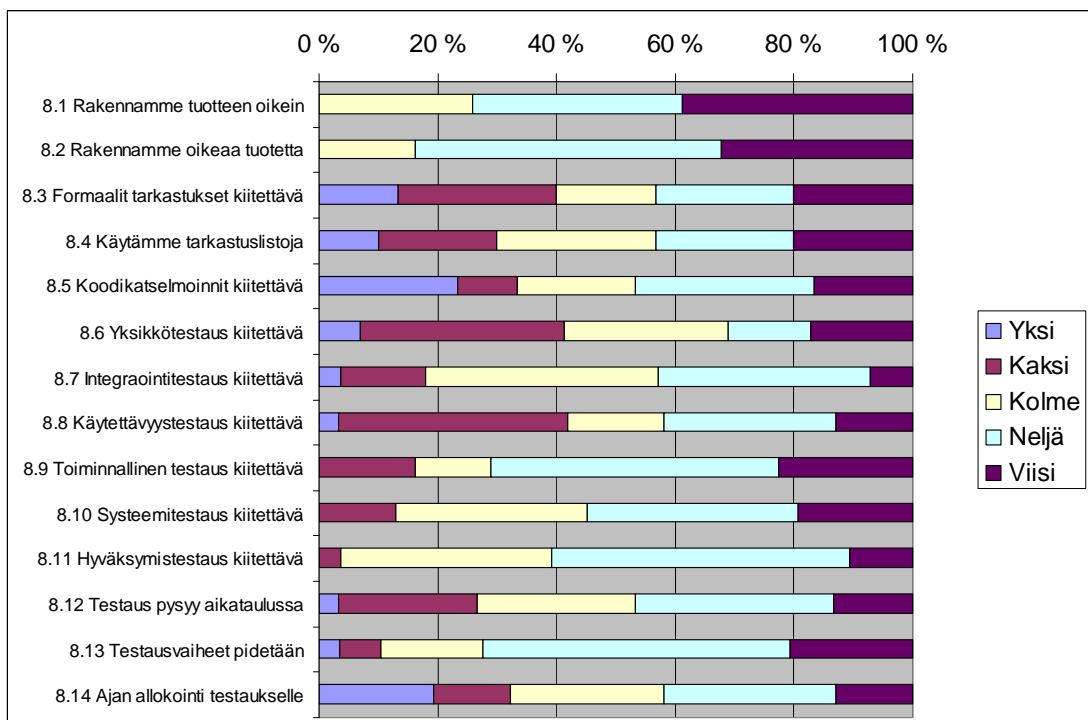
Tähän kysymykseen liittyen tarkistettiin ovatko vastaukset keskenään linjassa. Teoriassa kysymyksen 8.1 pitäisi olla sama kuin kysymysten 8.3, 8.4 ja 8.5 keskiarvo, koska molemmat mittaavat verifioivaa testausta. Jälkimmäisissä kysymyksissä kysytään sama asia purettuna osiin. Kuitenkin kysymysten 8.3, 8.4 ja 8.5 keskiarvo 3.1 on selvästi pienempi kuin kysymyksen 8.1 keskiarvo 4.2. Myös keskiarvoille tehty t-testi (paired t-test) osoittaa, että keskiarvot todella poikkeavat toisistaan, koska merkitsevyydeksi saatiin 0.000.

Vastaavasti kysymyksen 8.2 pitäisi olla sama kuin kysymysten 8.6, 8.7, 8.8, 8.9, 8.10 ja 8.11 keskiarvo, koska molemmat mittaavat validoivaa testausta. Myös tässä kysymysten keskiarvo 3.4 on selvästi pienempi kuin kysymyksen 8.2 keskiarvo 4.2. Myös tämän keskiarvoille tehty t-testi (paired t-test) osoittaa, että keskiarvot todella poikkeavat toisistaan, koska merkitsevyydeksi saatiin 0.000.

Luultavasti suuri ero johtuu siitä, että vastaajat eivät täysin ymmärtäneet mitä kysymykset 8.1 ja 8.2 tarkoittavat. On helppo sanoa, että ”rakennamme tuotteen oikein” tai ”rakennamme oikeaa tuotetta” miettimättä sen enempää mitä edellytyksiä väite pitää sisällään. Tutkimuksessa kysymys kysyttiin juuri niin kuin se on kirjoitettu, joten kysymysten yhteyttä validointiin ja verifiointiin ei välttämättä voinut huomata.



**Kuva 19. Ohjelmistotestaukseen liittyvien osakysymysten vastausten keskiarvot**



**Kuva 20. Ohjelmistotestaukseen liittyvien osakysymysten vastausten jakaumat**

Validoiva testaus näyttäisi olevan haastatelluissa organisaatioyksiköissä paremmalla tasolla kuin verifioiva testaus. Validoivasta testauksen osa-alueista edukseen erottuvat toiminnallinen – ja systeemitestaus. Lisäksi testausvaiheet näytettäisiin pidettävän eikä niitä leikata projektin deadlinesta riippumatta. Huonoimmalla tasolla organisaatioyksiköissä näyttäisi olevan yksikkötestaus, jonka keskiarvo jäi alle kolmen.

Kuten kappaleen alussa mainittiin, kysymys 8.1 mittaa verifiointia ja 8.2 validointia, joista muodostettiin summamuuttuja kuvaamaan testauksen tasoa (Testauksen taso 1). Lisäksi samaa asiaa kuvaamaan muodostettiin toinen testauksen summamuuttuja (Testauksen taso 2), joka koostuu kysymysten 8.3–8.11 keskiarvoista.

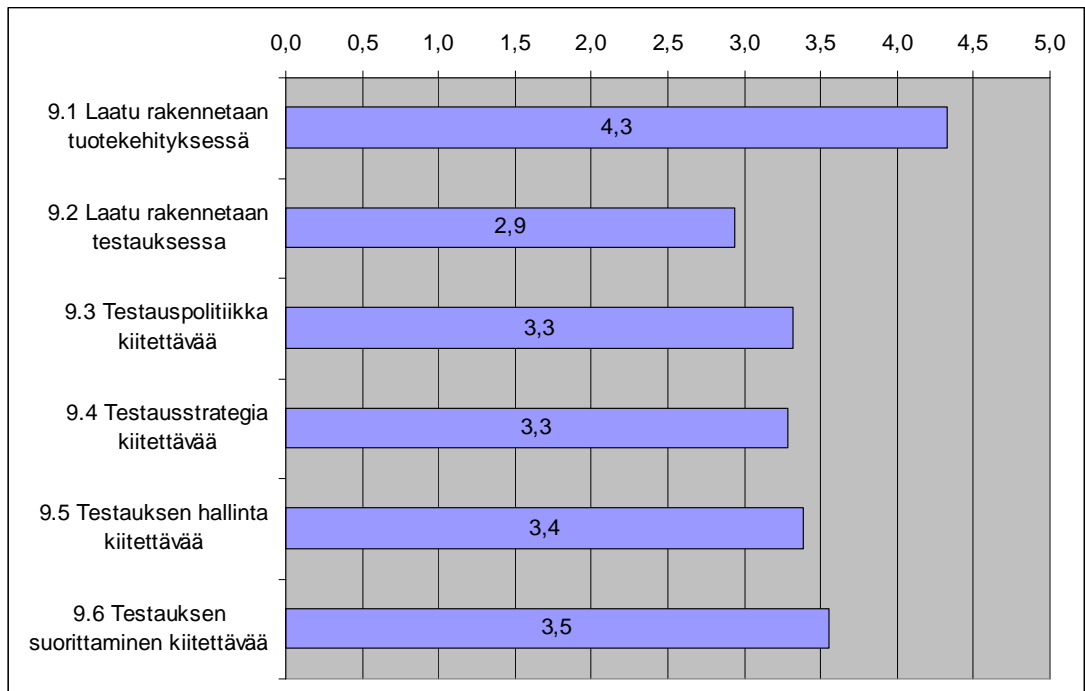
Muodostettujen muuttujien kaavat ovat:

$$\text{Testauksen taso 1} = (8.1+8.2)/2$$

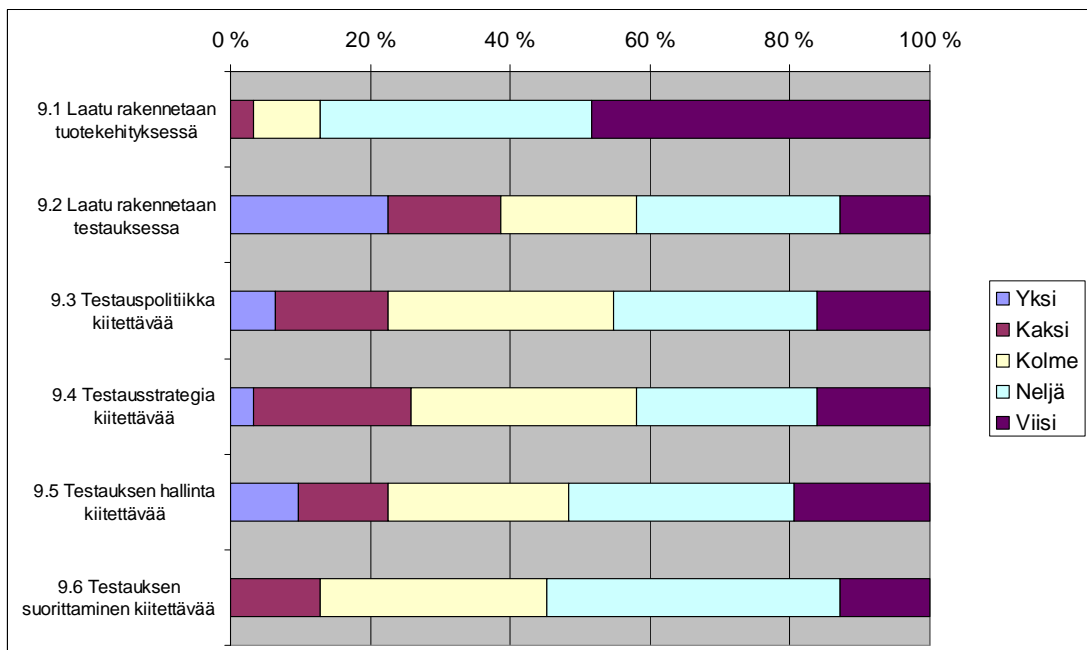
$$\text{Testauksen taso 2} = (8.3+8.4+8.5+8.6+8.7+8.8+8.9+8.10+8.11)/9$$

## **9.5 Standardin toteutuminen**

Standardin toteutumiseen liittyvät kysymykset ovat lomakkeen kysymykset 9.3 – 9.6. Haasteltavia pyydettiin arvioimaan uuden testausstandardin ISO/IEC 29119 sisältöä 5-portaisella Likert-asteikolla vastaamalla standardiin liittyviin väittämiin. Huomattavaa on, että uusi standardi ei vielä ole valmis eikä siten yritysten käytössä. Vastaus 1 merkitsi täysin eri mieltä ja 5 täysin samaa mieltä. Kuvassa 21 on esitetty osakysymysten keskiarvo ja kuvassa 22 osakysymysten jakauma.



**Kuva 21. ISO/IEC 29119 standardin toteutuminen – keskiarvot**



**Kuva 22. ISO/IEC 29119 standardin toteutuminen - jakaumat**

Vastaajat olivat pitkälle sitä mieltä, että laatu rakennetaan tuotekehityksessä. Sen sijaan laadun rakentuminen testauksen aikana keräsi keskiarvoksi hieman alle 3 eli keskimäärin vastaajat olivat hieman eri mieltä. Testausstrategian eri hierarkiatasot

olivat organisaatioyksiköissä keskimäärin suunnilleen samalla tasolla eli keskiarvo hieman yli kolmen.

Näistä kysymyksistä muodostettiin summamuuttuja, joka kuvaa standardin toteutumista yrityksessä.

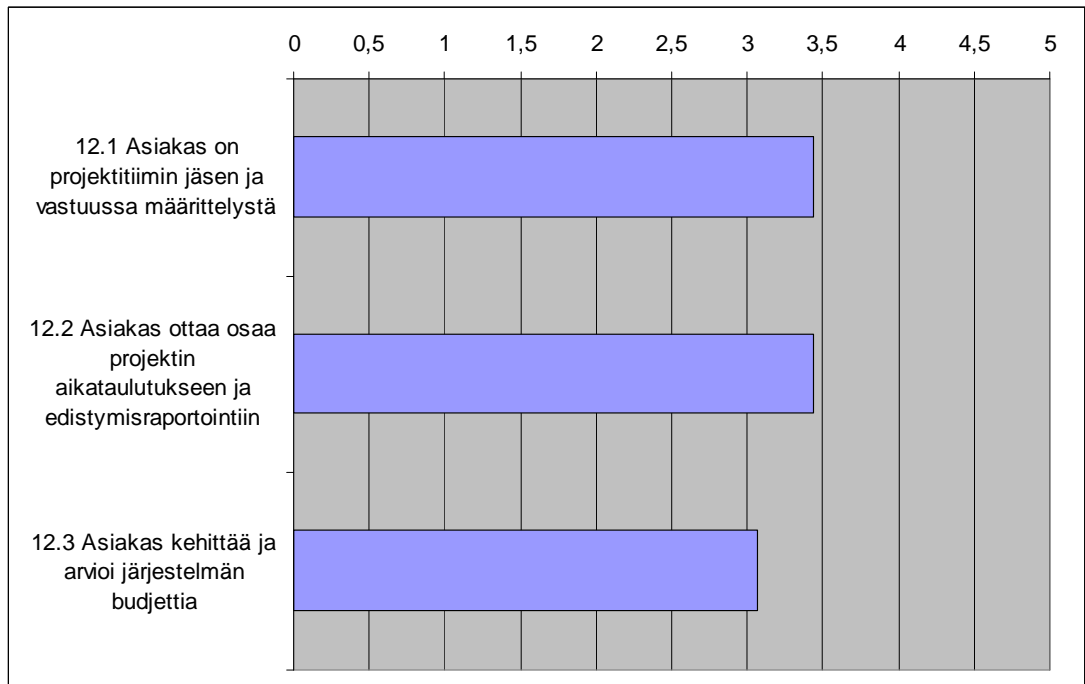
Muodostetun summamuuttujan kaava on:

$$\text{Standardin toteutuminen} = (9.3+9.4+9.5+9.6)/4$$

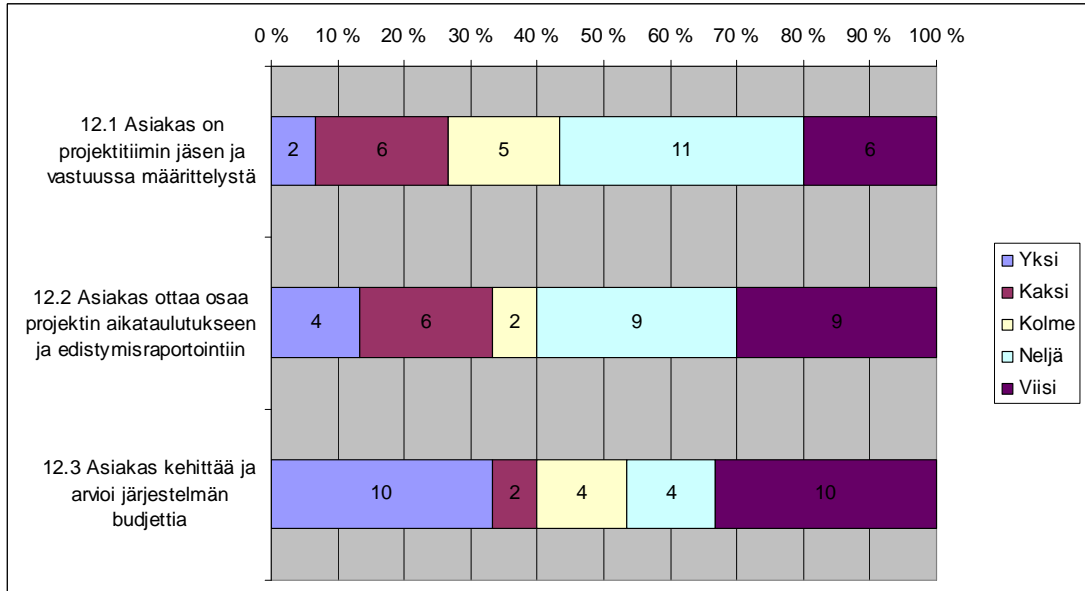
## **9.6 Asiakkaan osallistuminen**

Asiakkaan osallistumiseen liittyvät kysymykset ovat lomakkeen kysymykset 12.1–15.4. Kysymykset kuvaavat suunnitelmapohjaisen ohjelmistokehitysprosessin vaiheita siten, että kysymykset 12.1–12.3 liittyvät määrittelyyn, 13.1–13.4 suunnitteluun, 14.1–14.4 testaukseen ja kysymykset 15.1–15.4 ohjaukseen. Haasteltavia pyydettiin arvioimaan asiakkaan osallistumista määrittely -, toteutus -, testausvaiheessa sekä osallistumista yleiseen projektinhallintaan 5-portaisella Likert-asteikolla vastaamalla aihealueeseen liittyviin väittämiin. Vastaus 1 merkitsi täysin eri mieltä ja 5 täysin samaa mieltä. Kuvassa 23 on esitetty osakysymysten keskiarvo ja kuvassa 24 osakysymysten jakauma määrittelyvaiheessa.

Haastatelluissa organisaatioyksiköissä asiakas osallistuu keskimäärin melko aktiivisesti määrittelyvaiheessa. Asiakas osallistuu keskimäärin määrittelyyn projektitiimin jäsenenä sekä ottaa osaa projektin aikataulutukseen ja edistymisraportointiin aktiivisesti, molempien osakysymysten keskiarvo 3.4. Sen sijaan asiakas ei keskimäärin osallistu yhtä usein järjestelmän budjetin kehittämiseen ja arviointiin, keskiarvo 3.1. On hyvä huomioida kysymyksen 12.3 jakaumasta, että 10 haastatelluista yrityksistä asiakas ei osallistu budjetin kehittämiseen ja arviointiin.



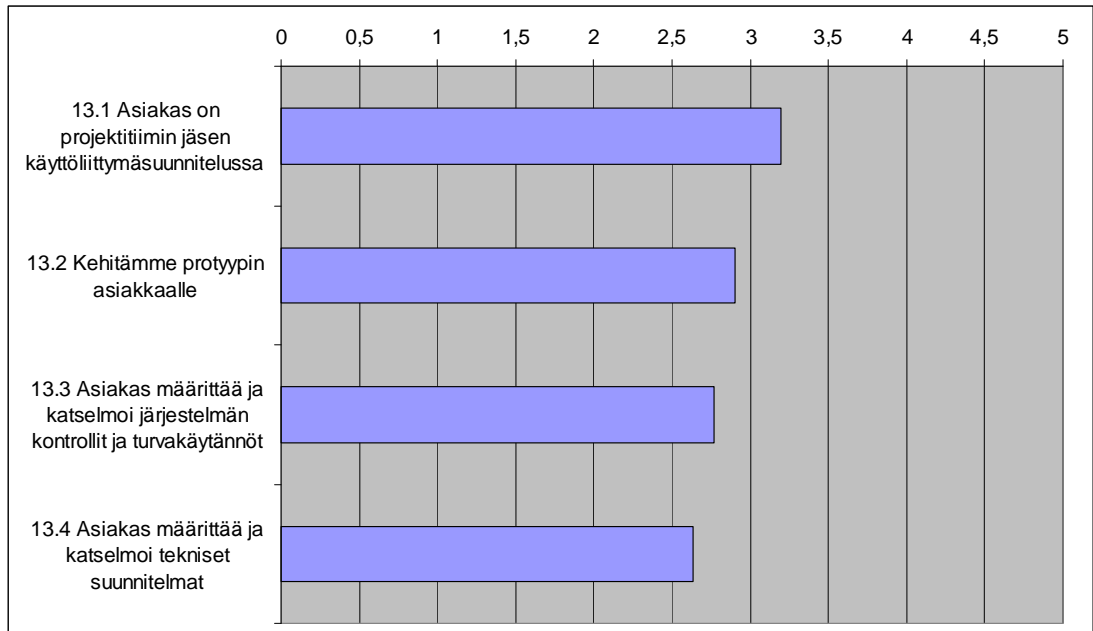
**Kuva 23. Tärkeimmän asiakkaan osallistumiseen määrittelyvaiheessa liittyvien osakysymysten vastausten keskiarvot**



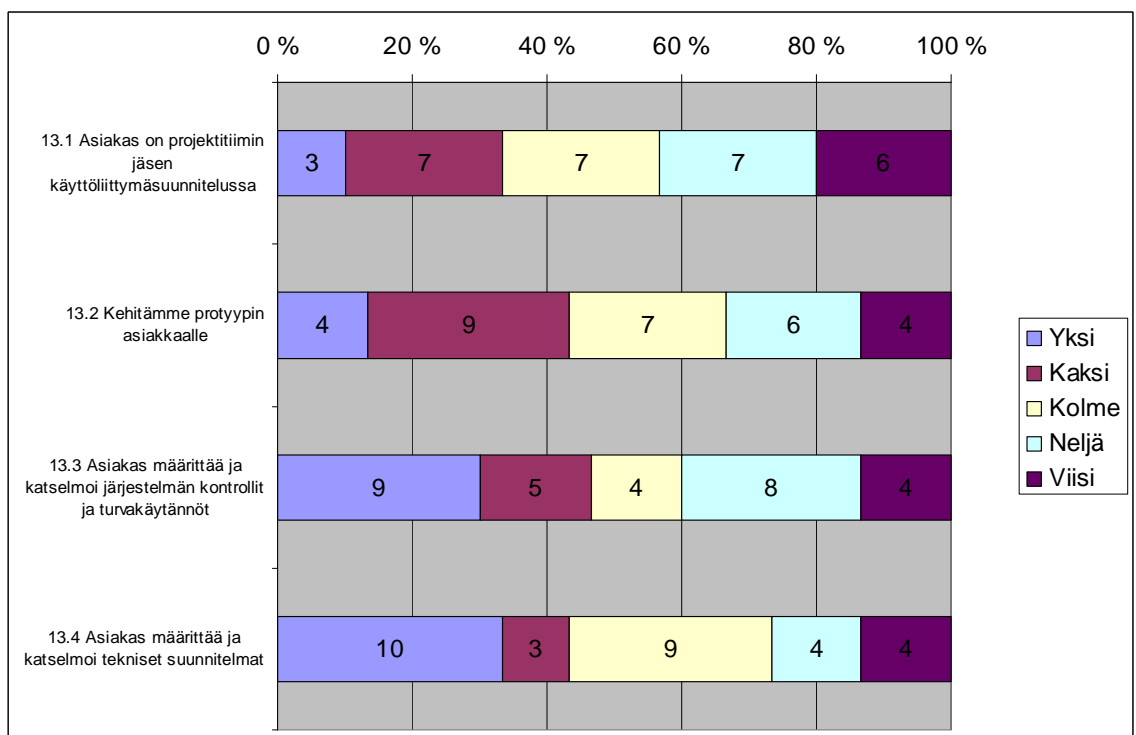
**Kuva 24. Tärkeimmän asiakkaan osallistumiseen määrittelyvaiheessa liittyvien osakysymysten jakaumat**

Suunnitteluvaiheessa haastatelluissa organisaatioyksiköissä asiakas osallistuu suunnitteluun huomattavasti vähemmän verrattuna määrittelyvaiheeseen, kuvat 25 ja 26.

Ainoastaan käyttöliittymäsuunnittelussa asiakas on melko aktiivisesti mukana, keskiarvo 3.2. Muiden suunnitteluvaiheen osakysymysten keskiarvot jäivät alle kolmen.

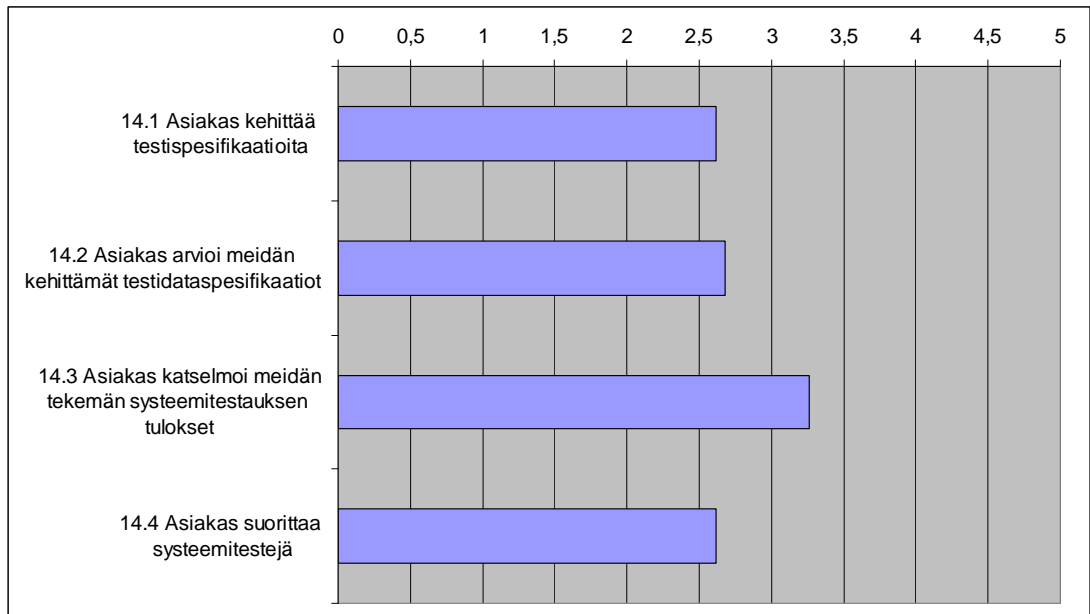


**Kuva 25. Tärkeimmän asiakkaan osallistumiseen suunnitteluvaiheessa liittyvien osakysymysten vastausten keskiarvot**

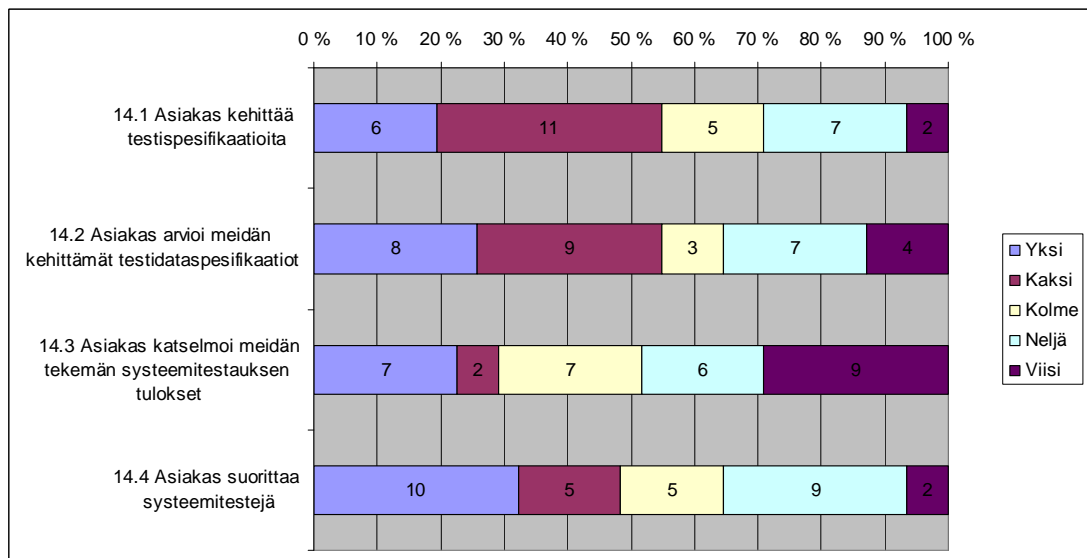


**Kuva 26. Tärkeimmän asiakkaan osallistumiseen suunnitteluvaiheessa liittyvien osakysymysten vastausten jakaumat**





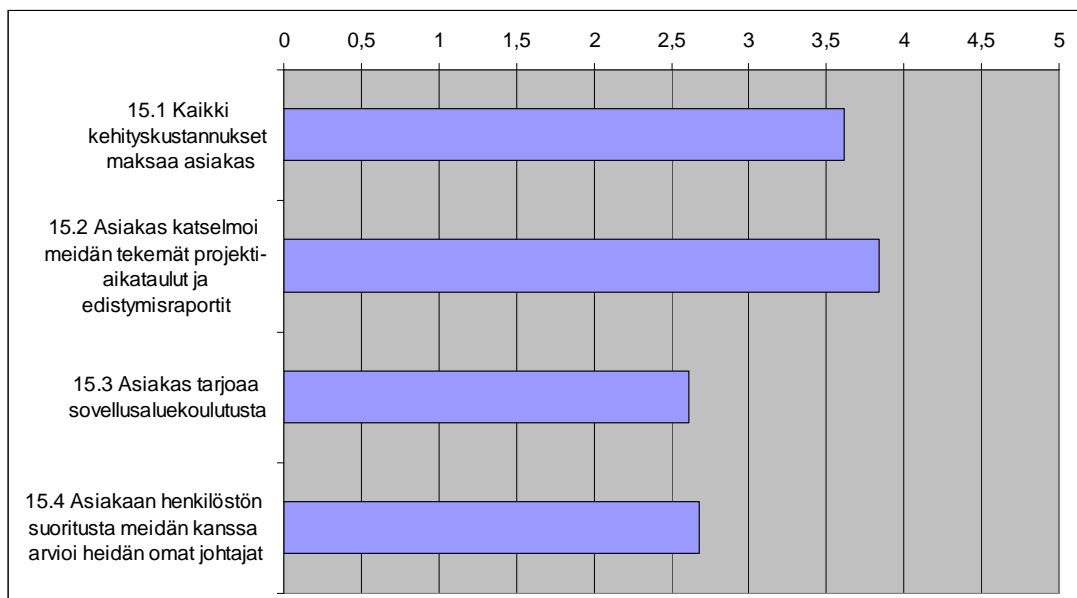
**Kuva 27. Tärkeimmän asiakkaan osallistumiseen testausvaiheessa liittyvien osakysymysten vastausten keskiarvot**



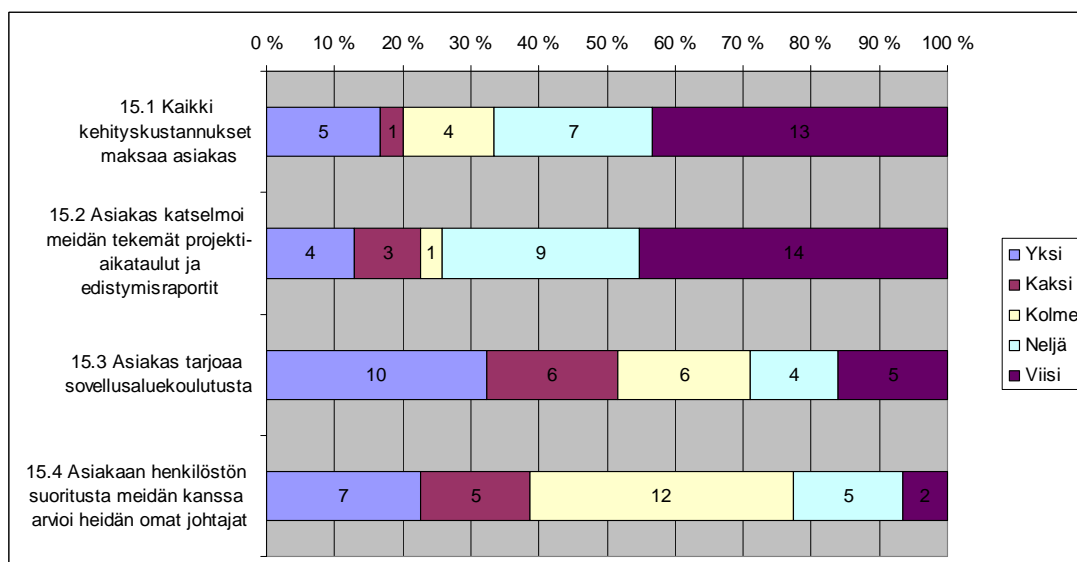
**Kuva 28. Tärkeimmän asiakkaan osallistumiseen testausvaiheessa liittyvien osakysymysten vastausten jakaumat**

Testausvaiheessa haastatelluissa organisaatioyksiköissä asiakas osallistuu huomattavasti vähemmän verrattuna sekä määrittelyvaiheeseen että suunnitteluvaiheeseen, kuvat 27 ja 28. Keskimäärin asiakas ainoastaan katselmoi vastaajarytymisen tekemän systeemitestauksen tulokset, keskiarvo 3.3. Sen sijaan asiakkaan osallistuminen testispesifikaatioiden kehittämiseen, osallistuminen testidataspesifikaation arviointiin ja

osallistuminen systeemitestien suorittamiseen ei keskimäärin ole kovin aktiivisella tasolla, osakysymysten keskiarvot hieman yli 2.5:n. Osakysymysten jakaumista näkee, että joissakin haastatelluissa organisaatioyksiköissä näihinkin asioihin asiakas osallistuu aktiivisesti.



**Kuva 29. Tärkeimmän asiakkaan osallistumiseen yleisesti projektiin liittyviin asioihin liittyvien osakysymysten vastausten keskiarvot**



**Kuva 30. Tärkeimmän asiakkaan osallistumiseen projektinohjaukseen liittyvien osakysymysten vastausten jakaumat**

Yleisesti projektiin liittyviin asioihin (kuvat 29 ja 30) haastatelluissa organisaatioyksiköissä asiakas osallistuu keskimäärin aktiivisesti kaikkien kehityskustannusten maksamiseen ja projekti aikataulujen sekä edistymisraporttien katselmointiin, molempien osakysymysten keskiarvo yli 3.5. Sen sijaan vähemmän aktiivisesti asiakas tarjoaa sovellusaluekoulutusta sekä asiakkaan henkilöstöä arvioidaan omien johtajien toimesta, molempien osakysymysten keskiarvo n. 2.5. Kysymyksestä 15.4 on hyvä huomata jakaumasta, kuinka paljon vastausta ”3” on annettu. Tämä viittaa siihen, että haastateltava ei tiennyt suoritetaanko tämän tyyppistä arviointia vai ei.

Kuhunkin vaiheeseen liittyville kysymyksille laskettiin Cronbachin alfakerroin (taulukko 7) ja muodostettiin vaihekohtaiset muuttujat. Projektinohjaukseen liittyvissä kysymyksissä havaittiin vain kysymysten 15.2 ja 15.3 osoittavan samaan suuntaan siten, että Cronbachin alfa-arvoksi saatiin yli 0.7.

**Taulukko 7. Cronbachin alfakerroin kysymyksille 12.1–15.4**

Asiakkaan osallistuminen		
Kysymykset	Muuttujien lukumäärä	Kysymysten välinen Cronbachin alfa
12.1, 12.2 ja 12.3	3	0.855
13.1, 13.2, 13.3 ja 13.4	4	0.772
14.1, 14.2, 14.3 ja 14.4	4	0.742
15.1, 15.2, 15.3 ja 15.4	4	0.599
15.2, 15.3 ja 15.4	3	0.669
15.2 ja 15.3	2	0.702

Vaihekohtaisten muuttujien kaavat ovat:

Asiakasosallistuminen määrittelyvaiheessa =  $(12.1+12.2+12.3)/3$

Asiakasosallistuminen suunnitteluvaiheessa =  $(13.1+13.2+13.3+13.4)/4$

Asiakasosallistuminen testausvaiheessa =  $(14.1+14.2+14.3+14.4)/4$

Asiakasosallistuminen ohjaukseen =  $(15.2+15.3)/2$

Vaihekohtaisista muuttujista muodostettiin summamuuttuja ”Asiakkaan osallistuminen”, joka summaa vaihekohtaiset muuttujan arvot ja näin ollen kuvaa asiakasosallis-

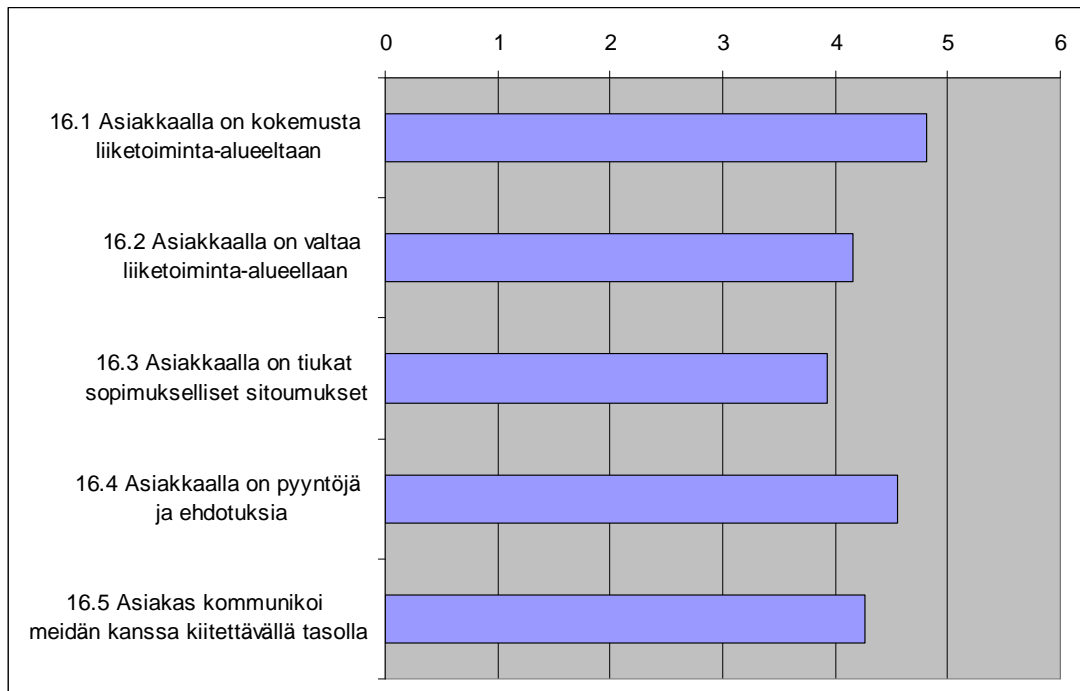
tumista kokonaisuudessaan. Muuttuja lisäksi skaalattiin asteikolle 0-5 jakamalla se neljällä.

Muodostetun summamuuttujan kaava on: Asiakkaan osallistuminen = (Asiakasosallistuminen määrittelyvaiheessa + Asiakasosallistuminen suunnitteluvaiheessa + Asiakasosallistuminen testausvaiheessa + Asiakasosallistuminen ohjaukseen)/4.

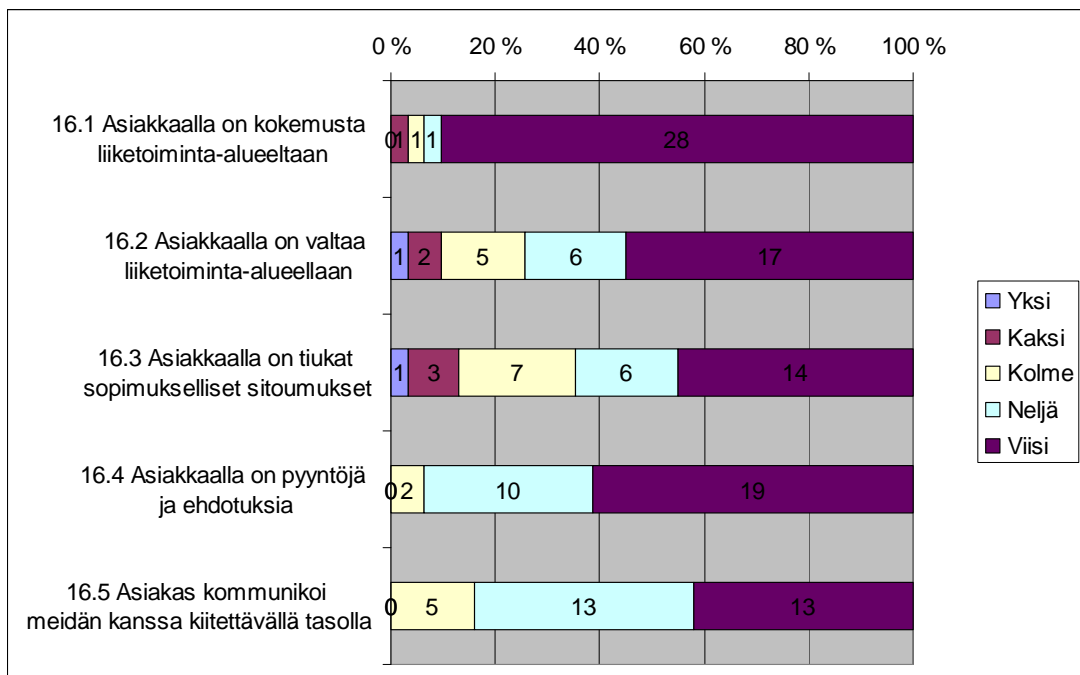
## **9.7 Asiakassuhde**

Asiakassuhteeseen liittyvät kysymykset ovat lomakkeen kysymykset 16.1-16.5. Haasteltavia pyydettiin arvioimaan asiakassuhteen tasoa 5-portaisella Likert-asteikolla vastaamalla aihealueeseen liittyviin väittämiin. Vastaus 1 merkitsi täysin eri mieltä ja 5 täysin samaa mieltä. Kuvassa 31 on esitetty osakysymysten keskiarvo ja kuvassa 32 osakysymysten jakauma.

Haastatelluissa organisaatioyksiköissä asiakkaalla on kokemusta liiketoiminta-alueeltaan. Tässä vastaukset olivat lähes yksimielisiä, sillä ainoastaan kolme yritystä vastasi jotain muuta kuin ”5” eli täysin samaa mieltä. Haastateltujen organisaatioyksiköiden asiakkaalla on myös pyyntöjä ja ehdotuksia. Ainoastaan kaksi organisaatioyksikköä vastasi neutraalilla ”3”, kun taas muut vastasi joko ”4” tai ”5”. Myös kolmen muun kysymyksen keskiarvot olivat noin neljä, mutta näissä hajonta oli jo huomattavasti suurempi. Suurimmalla osalla haastatelluista organisaatioyksiköistä asiakkaalla on valtaa liiketoiminta-alueellaan, asiakkaalla on tiukat sopimukselliset sitoumukset haastatellun organisaatioyksikön kanssa, ja asiakas kommunikoi haastatellun organisaatioyksikön kanssa kiitettävällä tasolla.



**Kuva 31. Asiakassuhteeseen liittyvien osakysymysten vastausten keskiarvot**



**Kuva 32. Asiakassuhteeseen liittyvien osakysymysten vastausten jakaumat**

Kysymyksistä vain kysymysten havaittiin 16.1 ja 16.2 osoittavan samaan suuntaan siten, että Cronbachin alfa-kertoimeksi saatiin yli 0.7, taulukko 8. Näistä kahdesta kysymyksestä muodostettiin muuttuja, joka kuvaa asiakassuhdetta.

Muodostetun muuttujan kaava on:

$$\text{Asiakassuhde} = (16.1+16.2)/2$$

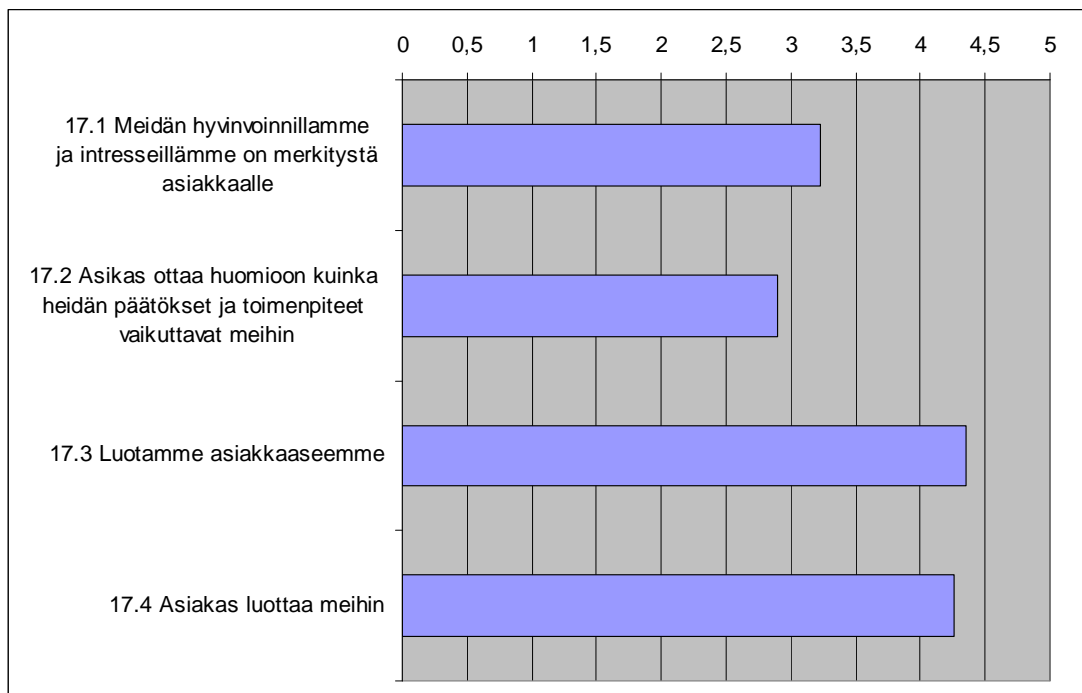
**Taulukko 8. Cronbachin alfa kerroin kysymyksille 16.1–16.5**

Asiakassuhde		
Kysymykset	Muuttujien lukumäärä	Kysymysten välinen Cronbachin alfa
16.1, 16.2, 16.3, 16.4 ja 16.5	5	0.547
16.1, 16.2, 16.3 ja 16.4	4	0.540
16.1, 16.2 ja 16.4	3	0,557
16.1 ja 16.2	2	0,739

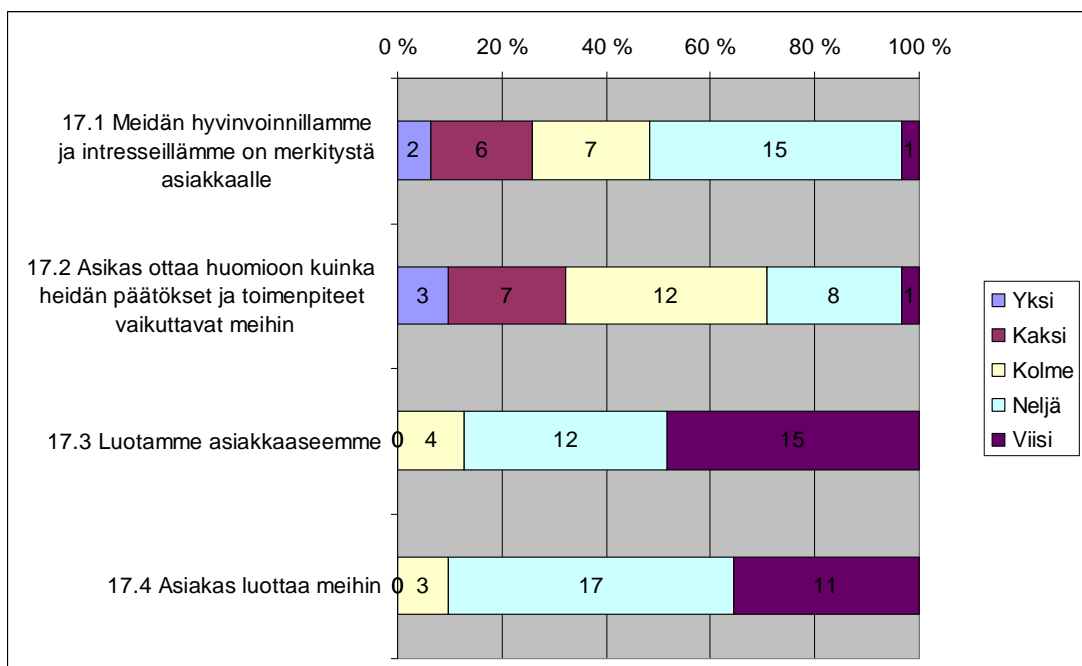
## **9.8 Luottamus**

Luottamukseen liittyvät kysymykset ovat lomakkeen kysymykset 17.1-17.4. Haasteltavia pyydettiin arvioimaan luottamuksen tasoa 5-portaisella Likert-asteikolla vastaamalla aihealueeseen liittyviin väittämiin. Vastaus 1 merkitsi täysin eri mieltä ja 5 täysin samaa mieltä. Kuvassa 33 on esitetty osakysymysten keskiarvo ja kuvassa 34 osakysymysten jakauma.

Haastatellut organisaatioyksiköt luottavat asiakkaaseensa ja asiakas luottaa heihin. Näissä vastaukset olivat lähes yksimielisiä, sillä ainoastaan muutama organisaatioyksikkö vastasi jotain muuta kuin ”4” tai ”5”. Sen sijaan osakysymykset liittyen haastatellun organisaatioyksikön hyvinvointiin ja intresseihin sekä asiakkaan päätösten toimenpiteiden vaikutus organisaatioyksikköön jakaumat ovat laajemmalle levinneitä keskiarvojen ollessa 3:n molemmin puolin.



**Kuva 33. Luottamukseen liittyvien osakysymysten vastausten keskiarvot**



**Kuva 34. Luottamukseen liittyvien osakysymysten vastausten jakaumat**

Kysymyksistä vain kysymysten 17.1 ja 17.2 havaittiin osoittavan samaan suuntaan siten, että Cronbachin alfa-kertoimeksi saatiin vähintään 0.7 (taulukko 9). Näistä kahdesta kysymyksestä muodostettiin muuttuja, joka kuvaa luottamusta.

Muodostetun muuttujan kaava on:

$$\text{Luottamus} = (17.1+17.2)/2$$

**Taulukko 9. Cronbachin alfa kerroin kysymyksille 17.1–17.4**

Kysymykset	Muuttujien lukumäärä	Kysymysten välinen Cronbachin alfa
17.1, 17.2, 17.3 ja 17.4	4	0.374
17.1, 17.2, 17.3	3	0.499
17.1 ja 17.2	2	0.699

## **9.9 Ohjelmiston laatu ja laatuattribuutit**

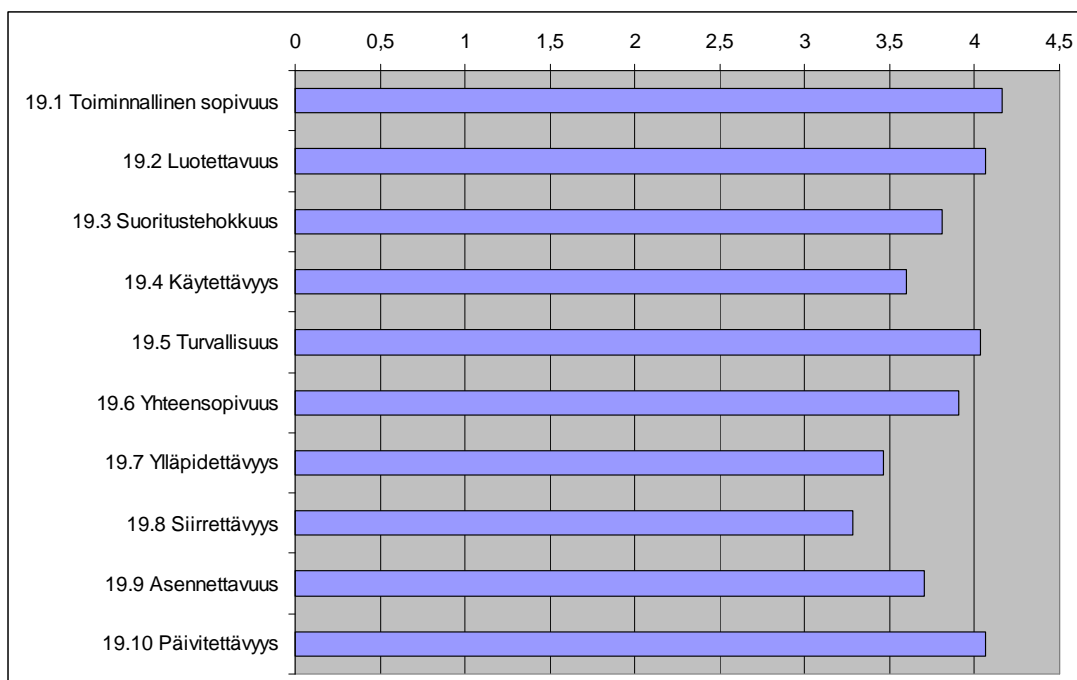
Laatuun liittyvät kysymykset ovat lomakkeen kysymykset 19.1-19.10. Haasteltavia pyydettiin arvioimaan ohjelmiston laatua vastaamalla osakysymyksiin ohjelmiston laatuattribuuteista 5-portaisella Likert-asteikolla. Lisäksi vastaajia pyydettiin arvioimaan yleisiin väittämiin laatuattribuuteista. Vastaus 1 merkitsi täysin eri mieltä ja 5 täysin samaa mieltä. Organisaatioyksikölle annettiin ohje jättää väittämä tyhjäksi, jos laatuattribuutti ei ollut organisaatioyksikölle merkityksellinen. Kuvassa 35 on esitetty osakysymysten keskiarvo ja kuvassa 36 osakysymysten jakauma.

Haastatelluissa organisaatioyksiköissä laatuattribuuttien keskiarvot olivat lähes kauttaaltaan hyvällä tasolla ja osakysymysten keskiarvot olivat yli 4. Hieman alemmalla tasolla olevia laatuattribuutteja olivat käytettävyys, ylläpidettävyys ja siirrettävyys, joiden keskiarvot olivat noin 3.5, siirrettävyyden vielä hieman senkin alle. Myös asennettavuuden jakaumassa näkyy useita ”1” vastauksia vaikka keskiarvo onkin melko korkea (3.7).

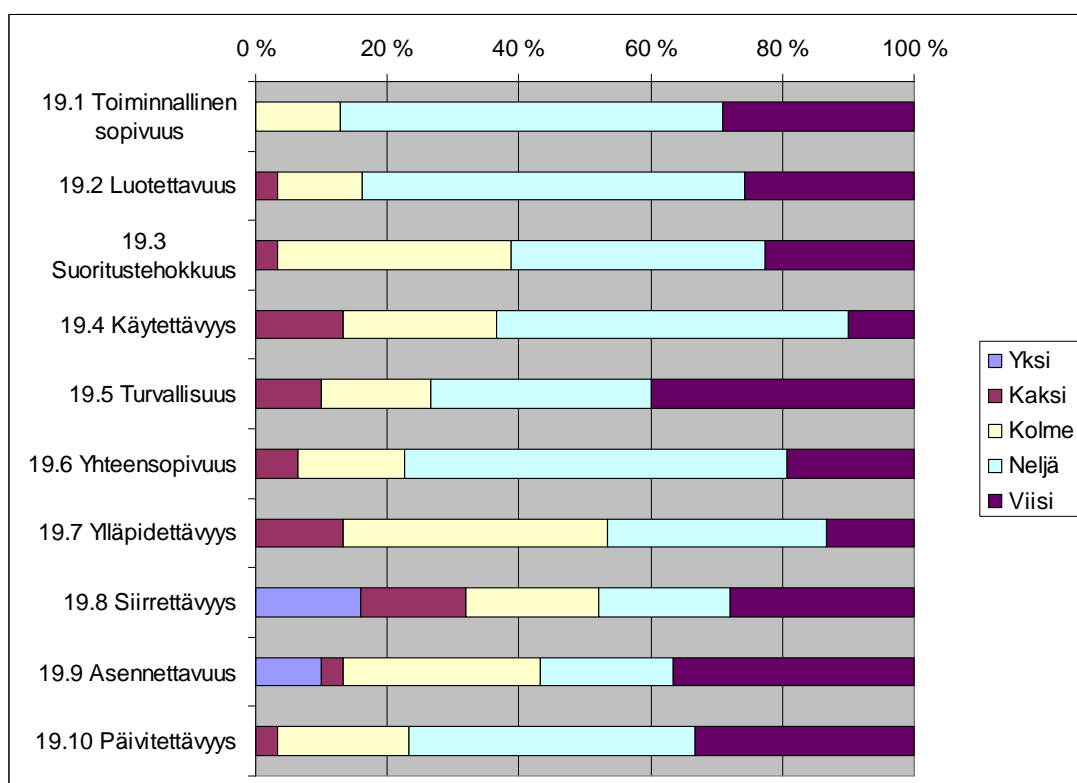
Keskiarvoja tulkittaessa on otettava huomioon, että laadulle asetetut tavoitteet ovat aina organisaatioyksikkö- tai jopa projektikohtaisia. Yksittäisen laatuattribuutin matala arvo tai heikko laatu voi jopa olla tavoite. Näin ollen esimerkiksi siirrettävyyden osalta ei voida suoraan vetää johtopäätöstä, että siirrettävyyttä pitäisi yleisesti ottaen



parantaa. Sen sijaan voidaan sanoa, että siirrettävyyttä ei ole laatuattribuuttina keskimäärin priorisoitu kovin korkealle.



**Kuva 35. Ohjelmiston laatuun liittyvien osakysymysten vastausten keskiarvot**



**Kuva 36. Ohjelmiston laatuun liittyvien osakysymysten vastausten jakaumat**

Laatuattribuuttien keskiarvosta muodostettiin laatua kuvaava muuttuja. Ainoastaan kysymyksiä 19.1-19.8 käytettiin Laatu-muuttujan muodostamiseen, koska ne ovat standardin 29119 mukaisia laatuattributteja. Kysymykset 19.9 ja 19.10 ovat TKK:n tutkijoiden kehittämiä laatuattributteja.

Muodostetun muuttujan kaava on:

$$\text{Laatu} = (19.1+19.2+19.3+19.4+19.5+19.6+19.7+19.8) / 8$$

Mikäli haastateltava henkilö ei vastannut kaikkiin laatua kuvaaviin väittämiin, jätettiin tyhjät väittämät huomioimatta laatu-muuttujaa laskettaessa, jolloin myös kaavan jakaja pieneni tyhjäksi jätettyjen kohtien lukumäärällä.

### **9.10 Korrelaatiomatriisi ja Kolmogorov-Smirnov**

Tulosten perusteella laskettiin korrelaatiomatriisi, jossa korrelaatiota laatu-muuttujan kanssa tarkasteltiin hypoteesien mukaisten muuttujien sekä muutaman muun tutkimusryhmän mielenkiintoiseksi katsoman muuttujan kanssa. Lisäksi asiakassuhteeseen liittyvien yksittäisten väittämien (väittämät 16.1–16.5) korrelaatio laadun kanssa laskettiin, koska esianalyysin perusteella asiakassuhteen summamuuttujan korrelaatio laadun kanssa näytti jäävän matalaksi.

Korrelaatiomatriisia varten laskettiin Kolmogorov-Smirnov matriisin muuttujille muuttujien normaalijakauman osoittamiseksi, taulukko 10. 0,05 riskitasoa käyttäessä Ohjelmistokehittäjien määrä, Ohjelmistokehitysmenetelmä ja Asiakassuhde sekä asiakassuhteeseen liittyvät yksittäisten väittämien muuttujat poikkeavat normaalijakaumasta. Sen sijaan 0,01 riskitasoa ainoastaan Ohjelmistokehittäjien määrä sekä asiakassuhteeseen liittyvistä väittämistä ”16.1 asiakkaallamme on kokemusta liiketoiminta-alueeltaan”, ”16.2 asiakkaallamme on valtaa liiketoiminta-alueellaan” sekä ”16.4 asiakkaallamme on pyyntöjä ja ehdotuksia” muuttujat poikkeavat normaalijakaumasta. Tulosten perusteella päätettiin 0,01 riskitasolla ei-normaalijakautuneet muuttujat pudottaa korrelaatiomatriisista.

**Taulukko 10. Kolmogorov-Smirnov korrelaatiomatriisin muuttujille**

<b>Muuttuja</b>	<b>Kolmogorov-Smirnov Z</b>
Ohjelmistokehittäjien määrä (ml. Testaajat)	0,002
Ketterien menetelmien käytön % -osuus	0,081
Olemassa olevien testausresurssien % -osuus tarpeesta	0,214
Tuotesuuntautuneisuus-%	0,322
Palvelusuuntautuneisuus-%	0,322
Ohjelmistokehitysmenetelmä	0,035
Kriittisyys	0,185
Testauksen taso 1	0,260
Testauksen taso 2	0,959
Standardin toteutuminen	0,703
Asiakasosallistuminen määrittelyvaiheessa	0,758
Asiakasosallistuminen suunnitteluvaiheessa	0,268
Asiakasosallistuminen testausvaiheessa	0,937
Asiakasosallistuminen ohjaukseen	0,630
Asiakkaan osallistuminen	0,927
Asiakassuhde	0,014
Väite 16.1: Asiakkaallamme on kokemusta liiketoiminta-alueeltaan	0,000
Väite 16.2: Asiakkaallamme on valtaa liiketoi- minta-alueellaan	0,004
Väite 16.3: Asiakkaallamme on tiukat sopi- mukselliset sitoumukset	0,023
Väite 16.4: Asiakkaallamme on pyyntöjä ja ehdotuksia	0,000
Väite 16.5: Asiakkaamme kommunikoi ja tekee yhteistyötä kiitettävällä tasolla meidän kanssa	0,026
Luottamus	0,064
Laatu	0,966

Kolmogorov-Smirnovin laskemisen jälkeen laskettiin muuttujista korrelaatiomatriisi. Taulukkoon 11 on koottu korrelaatiomatriisi tiivistettynä, jossa riippuvana muuttajana matriisissa on Laatu ja muut muuttajat ovat itsenäisiä muuttujia. Korrelaatiomatriisin perusteella laatu korreloi merkittävästi 99- % luottamustasoa käytettäessä

**Taulukko 11. Korrelaatiomatriisi. \*\* merkitsee 99% - ja \* 95% luottamustasoa**

<b>Itsenäinen muuttuja</b>	<b>Korrelaatiokerroin Laatu-muuttujan kanssa</b>
Ketterien menetelmien käytön %-osuus	-0,147
Olemassa olevien testausresurssien %-osuus tarpeesta	0,316
Tuotesuuntautuneisuus-%	-0,238
Palvelusuuntautuneisuus-%	0,238
Ohjelmistokehitysmenetelmä	-0,231
Kriittisyys	0,254
Testauksen taso 1	0,324
Testauksen taso 2	0,630 **
Standardin toteutuminen	0,622 **
Asiakasosallistuminen määrittelyvaiheessa	0,184
Asiakasosallistuminen suunnitteluvaiheessa	0,366 *
Asiakasosallistuminen testausvaiheessa	0,190
Asiakasosallistuminen ohjaukseen	0,376 *
Asiakkaan osallistuminen	0,242
Asiakassuhde	-0,072
Väite 16.3: Asiakkaallamme on tiukat sopimuskelliset sitoumukset	0,325
Väite 16.5: Asiakkaamme kommunikoi ja tekee yhteistyötä kiitettävällä tasolla meidän kanssa	0,416*
Luottamus	0,537**

Testauksen taso 2:n, Standardin toteutumisen ja Luottamuksen kanssa. 95- % luottamustasoa käytettäessä edellisten lisäksi laatu korreloi myös Asiakasosallistumisen suunnitteluvaiheessa, Asiakasosallistumisen ohjaukseen sekä asiakassuhteeseen liittyvistä kysymyksistä kysymyksen ”16.5 Asiakkaamme kommunikoi ja tekee yhteistyötä kiitettävällä tasolla meidän kanssa” – muuttujien kanssa.

### **9.11 Lineaarinen regressiomalli**

Merkitsevien korrelaatioiden perusteella muodostettiin lineaarinen regressiomalli käyttämällä SPSS ohjelmassa Step- sekä taaksepäin eliminointi-metodia. Step-metodissa malliin lisätään ja poistetaan muuttujia yksi kerrallaan tietyn algoritmin

mukaisesti, mikäli ne ovat tilastollisesti merkitseviä. Taaksepäin eliminointi -metodissa mallia testataan ensin kaikilla selittävillä muuttujilla ja mallista tiputetaan yksi muuttuja kerrallaan, mikäli muuttuja ei ole tilastollisesti merkitsevä.

Lineaarisessa regressiomallissa selittäviä muuttujia ovat Testauksen taso 2, Standardin toteutuminen, Asiakasosallistuminen suunnitteluvaiheessa, Asiakasosallistuminen ohjaukseen, Luottamus ja asiakassuhteeseen liittyvistä väittämistä väite 16.5: ”Asiakkaamme kommunikoi ja tekee yhteistyötä kiitettävällä tasolla meidän kanssa”. Selittävistä muuttujista riippuvana muuttujana on laatu.

Lineaarisen regressionmallin oletuksena on, että muuttujat ovat lineaarisesti riippuvaisia toisistaan. Oletuksen varmistamiseksi muuttujille piirrettiin sirontakuviot, jossa Y-akselilla on esitetty laatu ja X-akselilla selittävä muuttuja (liite 2). Lineaarisuusoletuksen mukaan pisteiden pitäisi sijoittua tasaisesti kuvioon siten, että niiden läpi voi piirtää lineaarisen viivan, jolloin pisteitä sijoittuu viivan molemmille puolille yhtä paljon. Kaikilla selittävillä muuttujilla oletus osoittautui todeksi lukuun ottamatta asiakassuhteeseen liittyvää väittämää 16.5: ”Asiakkaamme kommunikoi ja tekee yhteistyötä kiitettävällä tasolla meidän kanssa”, jolla pisteet muodostivat 3 selkeästi erillistä aluetta kuviossa. Tästä syystä se päätettiin pudottaa regressiomallin laskennasta pois.

Analyysin perusteella parhaiten varianssin selittäväksi saatiin kaksi mallia, joista toinen 0.05 ja toinen 0.01 riskitasolla. Riskitason valinta vaikuttaa siihen, kuinka alhainen t-testin merkitsevyys mallin kertoimille ja vakiolle vaaditaan, jotta H0 hypoteesi voidaan hylätä. Hypoteesit ovat:

H0: muuttajan kerroin ei poikkea nolasta

H1: muuttujan kerroin poikkeaa nolasta

0.01 riskitasolla tilastollisesti merkittäväksi malliksi saatiin seuraavanlainen malli:

Laatu = 2.407 + 0.433 \* Testauksen taso

Mallissa selittävät muuttujat selittävät mallin varianssista 35,2 %, taulukko 12. Lukema on pieni ja suurin osa varianssin selityksestä jääkin mallin ulkopuolelle. Taulukossa 12 on myös esitetty F-testin merkitsevyys regressioyhtälön kertoimille. Testissä tavoitellaan mahdollisimman pientä lukua, jotta H0 hypoteesi voidaan hylätä. Hypoteesit ovat:

H0: kaikki regression kertoimet ovat nollia

H1: regression kertoimet poikkeavat nolasta

Malli on siis tilastollisesti merkitsevä, sillä F-testin merkitsevyys on 0.001 joka on pienempi kuin 0.01, taulukko 13. Tällöin H0 hypoteesi voidaan hylätä riskitasolla 0.01.

Koska mallin muuttujien selitysaste varianssista on vain suunnilleen kolmasosa, tarkoittaa se sitä, että mallin selitysaste on varsin huono. Näin ollen päätettiin tehdä uusi malli tehdä käyttämällä riskitasoa 0.05, jolloin malliksi saatiin seuraavanlainen yhtälö (taulukko 16):

Laatu = 1.838 + 0.271 \* Standardin toteutuminen + 0.163 \* Asiakasosallistuminen suunnitteluvaiheessa + 0.194 \* Luottamus

**Taulukko 12: Regressioyhtälön korrelaatiokerroin ja korrelaationeliö riskitasolla 0.01**

Malli	R	R <sup>2</sup>	Estimaatin keskivirhe
Laatu	,594 <sup>a</sup>	,352	,4484627
a. Ennustajat: (Vakio), Testauksen taso 2			

**Taulukko 13. Varianssianalyysi<sup>b</sup> riskitasolla 0.01**

Malli		Neliöiden summa	df	F	Merkitsevyys
Laatu	Regression	3,065	1	15,238	,001 <sup>a</sup>
	Residual	5,631	28		
	Total	8,696	29		
a. Ennustajat: (Vakio), Testaus2					
b. Riippuva muuttuja: Laatu					

Mallissa selittävät muuttujat selittävät selitettävän muuttujan vaihtelusta 54 %, taulukko 14. Lukema on jo huomattavasti suurempi, mutta selvää on, että malliin kaivattaisiin lisää selittäviä muuttujia lopun 46 % täyttämiseksi. Myös varianssianalyysi antoi regressiomallille merkitsevyydeksi 0.000 eli malli on tilastollisesti merkitsevä riskitasolla 0.01 ( $0.000 < 0.01$ ), taulukko 15.

Lineaarissa regressiossa oletetaan, että mallin muuttujat ovat keskenään toisistaan riippumattomia. Yleisenä nyrkkisääntönä tilastotieteissä on, että toleranssin pitäisi olla enemmän kuin 0.20, jotta muuttujat voidaan olettaa riippumattomiksi. Jokaisen mallin muuttujan toleranssi on selvästi tämän yli, joten ne ovat toisistaan riippumattomia.

Mallissa huomio kiinnittyy siihen, että Testauksen taso 2 muuttujaa on pudonnut mallista pois vaikka 0.01 riskitasoa käytettäessä se oli ainoa selittävä muuttuja. Testauksen taso 2 muuttujan ollessa mallissa selitysasteeksi saataisiin hieman parempi 55,9 %. Sen sijaan kertoimien t-testissä merkitsevyydeksi saataisiin 0,304 mikä on huomattavasti suurempi kuin käytetty riskitaso 0.05. Näin ollen valitulla riskitasolla ei Testauksen taso 2 muuttujaa voi ottaa malliin.

Mallin selittävästä varianssista 27,1 % selittää standardin toteutuminen, 16,3 % asiakasosallistuminen suunnitteluvaiheessa sekä 19,4 % luottamus. Kaikilla muuttujilla on positiivinen kerroin, kuten hypoteesien teoreettisissa perusteluissa oletettiin. Näin ollen tämän tutkimuksen perusteella standardin toteutumisen, luottamuksen ja asiakasosallistumisen suunnitteluvaiheessa lisääminen vaikuttaa positiivisesti laatuun.

**Taulukko 14. Regressioyhtälön korrelaatiokerroin ja korrelaationeliö riskitasolla 0.05**

<b>R</b>	<b>R<sup>2</sup></b>	<b>Estimaatin keskivirhe</b>
,735 <sup>c</sup>	,540	,3922876
c. Ennustajat: (Vakio), „Luottamus, Asiakasosallistuminen suunnitteluvaiheessa, Standardin toteutuminen		

**Taulukko 15. Varianssianalyysi<sup>c</sup>, riskitaso 0.05**

Malli	Neliöiden summa	Df	F	Merkitsevyys
Regression	4,695	3	10,169	,000 <sup>c</sup>
Residual	4,001	26		
Total	8,696	29		

c. Ennustajat: (Vakio), Asiakkaamme kommunikoi ja tekee yhteistyötä kiitettävällä tasolla meidän kanssa, Asiakasosallistuminen suunnitteluvaiheessa, Luottamus, Standardin toteutuminen  
e. Riippuva muuttuja: Laatu

**Taulukko 16. Kertoimet<sup>a</sup>, riskitaso 0.05**

Malli	Standardoimattomat kertoimet		T	Merkitsevyys	Colineearisuus
	B	Keskivirhe			Toleranssi
(Vakio)	1,838	,365	5,042	,000	
Standardin toteutuminen	,271	,084	3,213	,003	,874
Asiakasosallistuminen suunnitteluvaiheessa	,163	,073	2,224	,035	,985
Luottamus	,194	,088	2,207	,036	,865

a. Riippuva muuttuja: Laatu



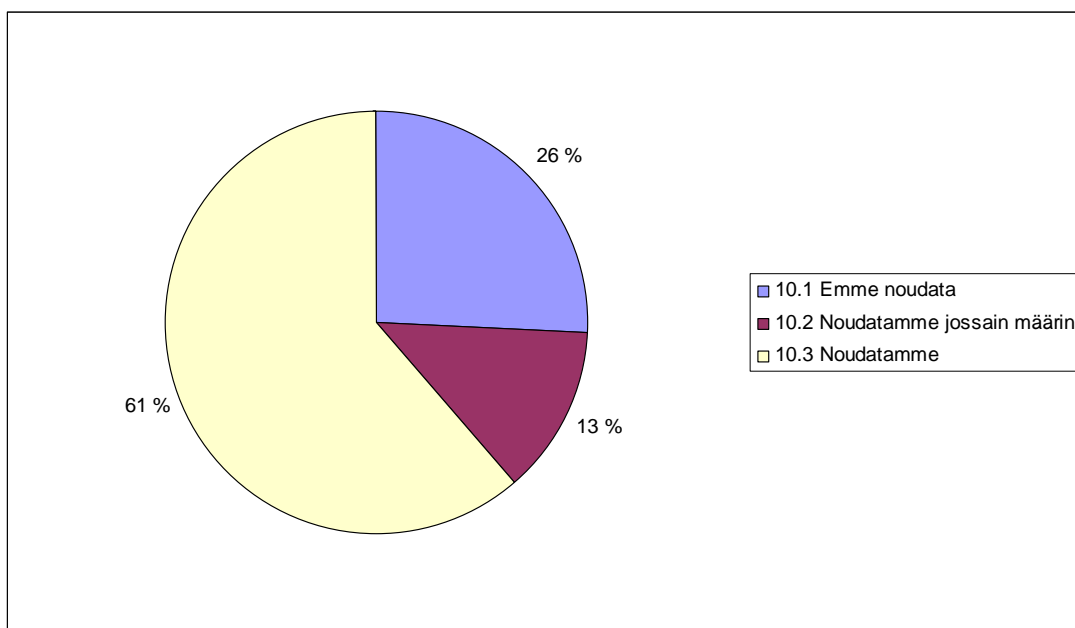
## 10 KARTOITTAVIEN KYSYMYSTEN TULOKSET

Kartoittavilla kysymyksillä selvitettiin, millaisia ohjelmistokehitykseen liittyviä käytäntöjä, menetelmiä ja työkaluja organisaatioyksiköissä käytetään. Lisäksi kartoitettiin ohjelmistotestaukseen liittyviä ongelmia ja parannusehdotuksia, merkittävimpiä tapoja miten asiakas voi vaikuttaa ohjelmiston laatuun sekä suurimpia hyötyjä ja haittoja ohjelmistokehityksen tai testauksen ulkoistamisessa.

Seuraavissa kappaleissa on esitetty kartoittavien kysymysten jakaumat. Kartoittavien kysymysten tuloksia käytetään hyväksi myös MASTO-projektin kvalitatiivisessa tutkimusosassa, jotta säästetään datankeruun kustannuksissa.

### 10.1 Systemaattinen menetelmä tai prosessi

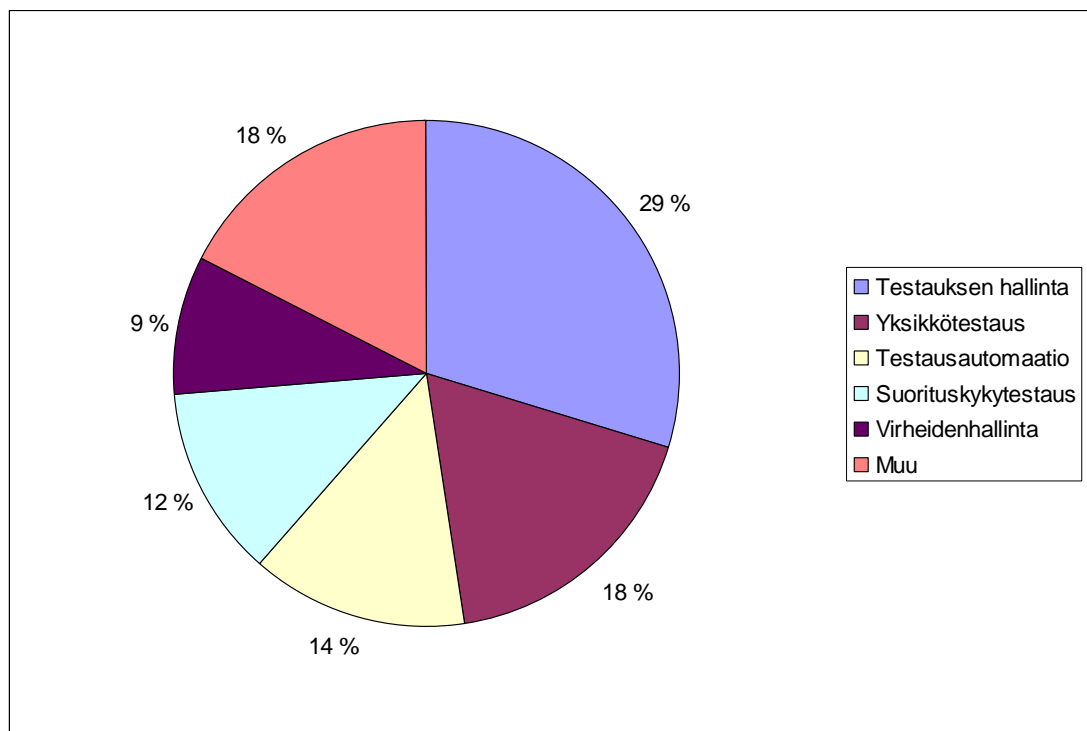
Haastatelluista organisaatioyksiköistä 61 % noudatti, 13 % noudatti jossain määrin ja 26 % ei noudattanut lainkaan systemaattista menetelmää tai prosessia, kuva 37. Organisaatioyksiköistä joissa prosessia noudatettiin, käytettiin prosessina itse määriteltyä prosessia. Oman prosessin taustalla olivat käytännöt muun muassa RUP, TPI (Total Process Improvement) ja XP.



Kuva 37. Systemaattisen menetelmän tai prosessin soveltaminen

## 10.2 Käytetyt testaustyökalut

Kuvassa 38 on esitetty käytettyjen testaustyökalujen jakauma. Haastatelluissa organisaatioyksiköissä eniten käytössä olivat testauksenhallinnan työkalut 29 %, joista eniten yksittäisiä mainintoja keräsivät Mercuryn (nykyisin HP:n) QualityCenter ja TestDirector sekä Microsoftin Excel. Muutoin mainitut työkalut jakautuivat tasaisesti eri valmistajien ja tuotemerkkien kesken. On hyvä huomata, että tutkimuksessa ei ole vertailtu eri valmistajien tai tuotenimien keskinäistä paremmuutta.



Kuva 38. Käytettyjen testaustyökalujen jakauma

Toiseksi eniten mainintojen keräsi yksikkötestaukseen tarkoitetut työkalut (18 %). Tähän luokkaan liittyvistä ei-itsetehdyistä työkaluista mainittiin xUnit (Junit, Nunit jne.), CPPUnit ja SimpleTest.

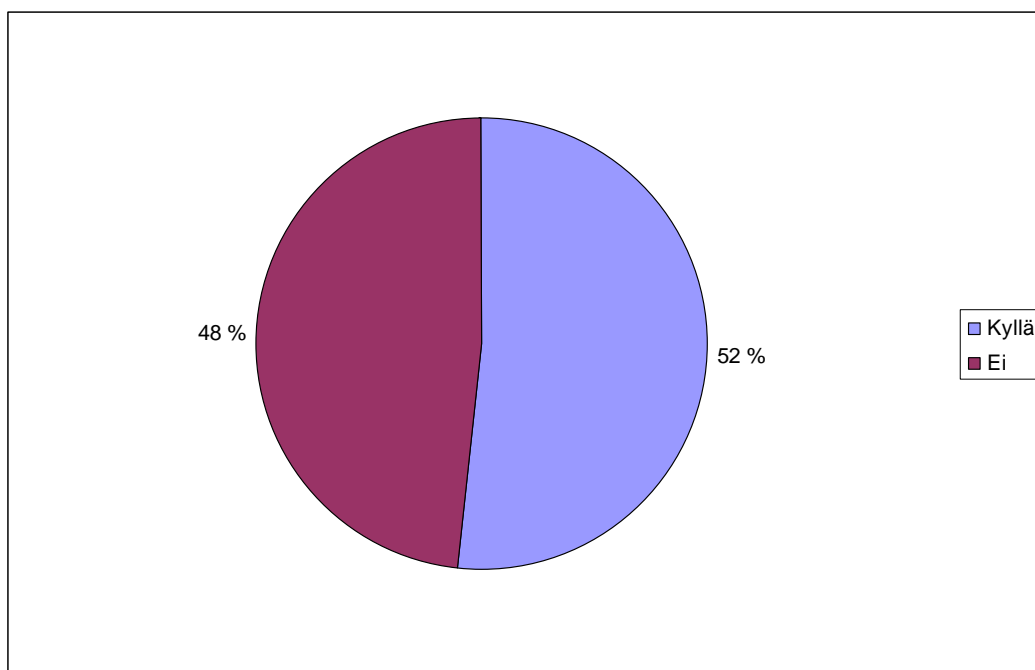
Testausautomaatio-luokkaan (14 %) liittyvistä työkaluista yksittäisiä mainintoja keräsivät muun muassa Rational functional tester, TestComplete ja OpenTTCN. Suorituskyky-luokkaan (12 %) liittyvissä työkaluissa mainintoja keräsivät Rational Performance tester, QALoad, Mercury (HP) LoadRunner, Jmeter ja VantageAnalyzer.

Virheiden-hallinta luokkaan (9 %) liittyvistä työkaluista mainintoja keräsivät muun muassa TrackRecord ja TestTrack.

Koska testaustyökaluja mainittiin hyvin monenlaisia, on mukaan otettu myös luokka ”Muu” (18 %), johon kuuluvat muun muassa sähkötekniset mittauslaitteet, tietojen-jako-, koodin analysointi- ja projektinhallintatyökalut. Näistä esimerkiksi projektin-hallintatyökaluista mainittiin Microsoftin Project.

### 10.3 Laatu järjestelmäsertifikaatti tai kypsyystasoluokitus

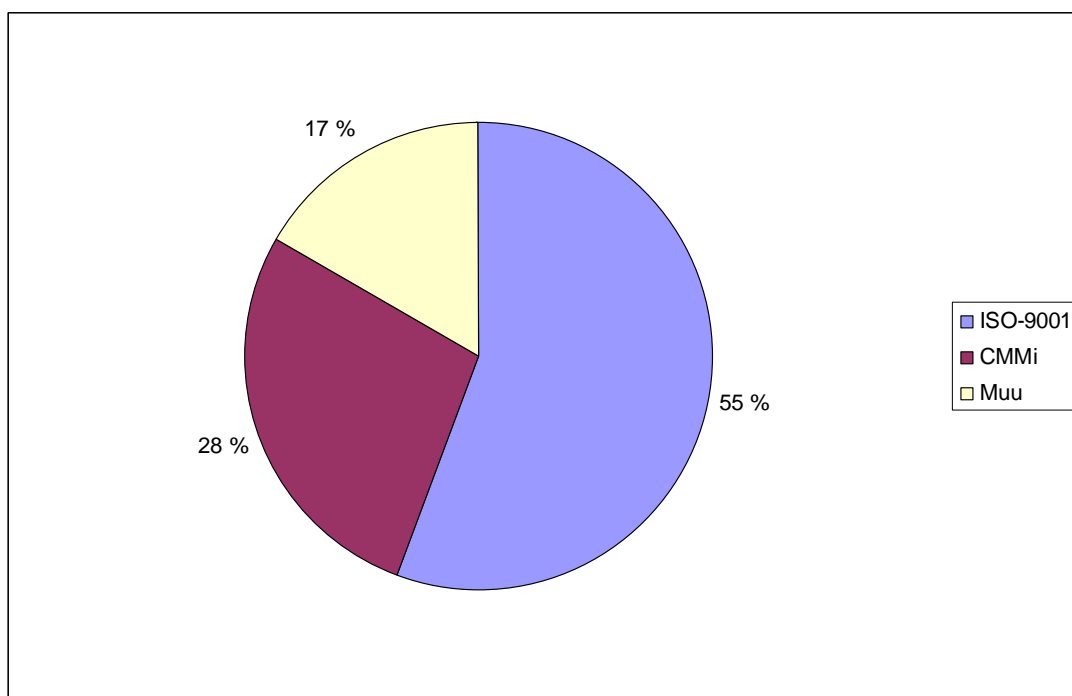
Vastanneista hieman yli puolella (52 %) on olemassa joko laatu järjestelmäsertifikaatti tai kypsyystasoluokitus, kuva 39.



**Kuva 39.** Yrityksellä laatu järjestelmäsertifikaatti tai kypsyystasoluokitus

Kuvassa 40 on esitetty käytössä olevien laatu järjestelmien tai kypsyystasoluokitusten jakauma. Haastatelluissa organisaatioyksiköissä yleisin käytössä oleva laatu järjestelmäsertifikaatti on ISO-9001 (55 %). Myös kypsyystasoluokitus CMMi (28 %) on yleisesti käytössä. Kategoriaan muu (17 %) kuuluu kolme yksittäistä vastausta, jotka ovat SPICE, Suomen laatu palkinto sekä oma luokitus. On hyvä huomata, että järjes-

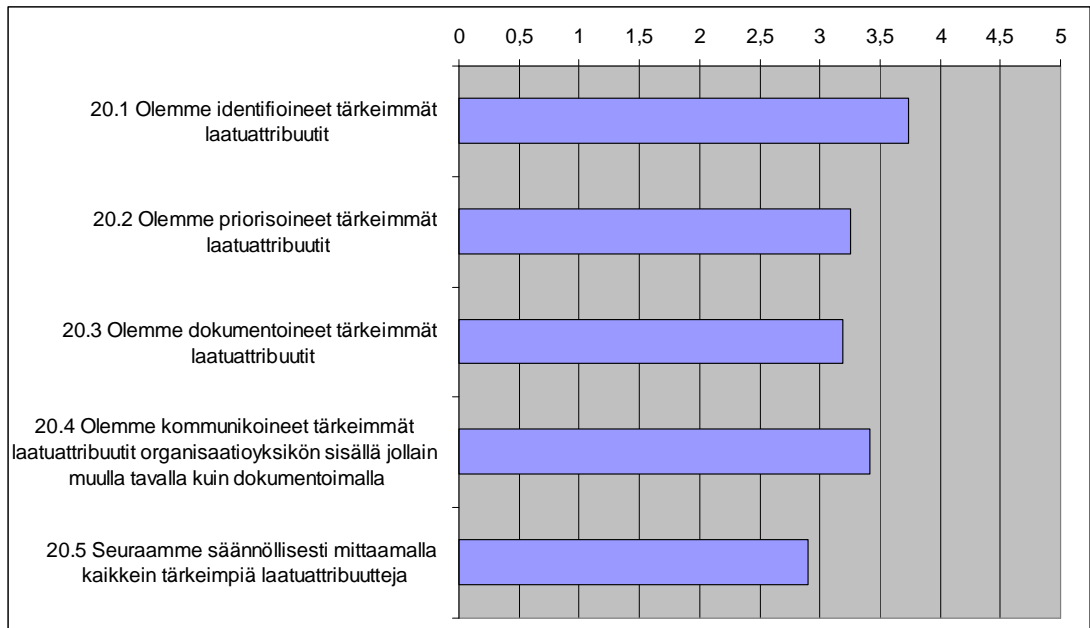
telmät eivät ole toisensa poissulkevia ja osalla vastanneista organisaatioyksiköistä järjestelmiä oli käytössä useampia kuin yksi.



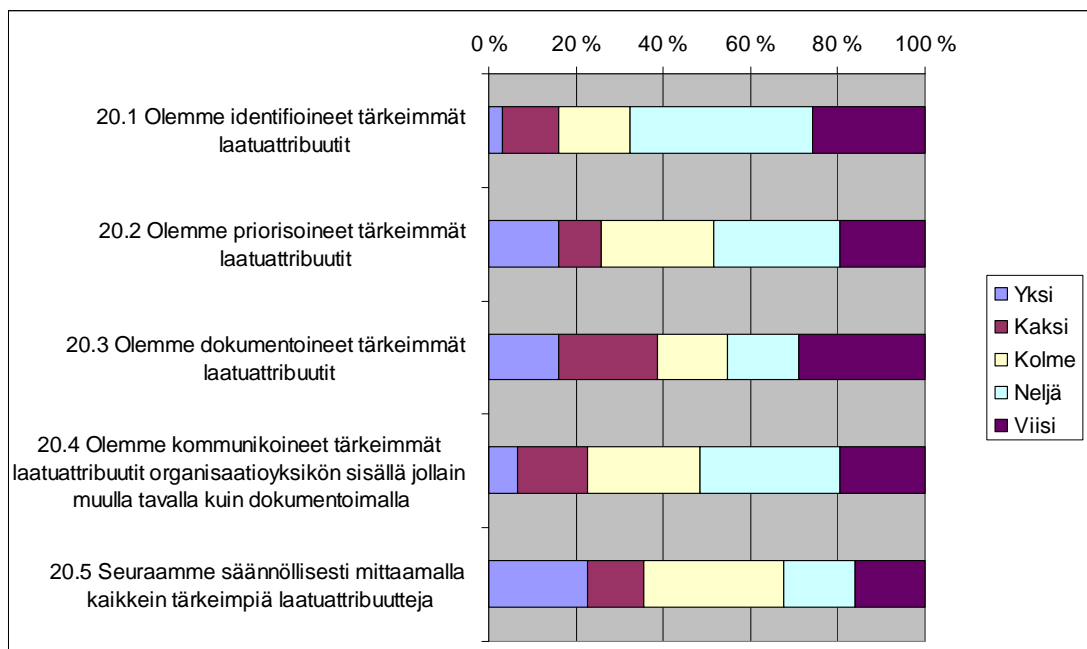
Kuva 40. Käytössä olevat laatuajärjestelmät tai kypsyystasoluokitukset

#### **10.4 Yleisiin laatuattributteihin liittyvät kysymykset**

Yleisiin laatuattributteihin liittyvistä kysymyksistä (kuvat 41 ja 42) laatuattribuutit oli identifioitu haastatelluissa organisaatioyksiköissä melko usein, koska 2/3 haastateluista vastasi tähän ”4” tai ”5” ja keskiarvo on noin 3.7. Sen sijaan muissa osakysymyksissä vastausten painoarvo on enemmän vasemmalla (lähempänä ykköstä). Erityisesti tärkeimpien laatuattribuuttien säännöllistä mittausta ei monissa yrityksissä tehdä, koska vastauksia ”1” annettiin yli 20 % vastauksista ja keskiarvokin jäi alle kolmen.



**Kuva 41. Laatuattributteihin liittyvien osakysymysten vastausten keskiarvot**

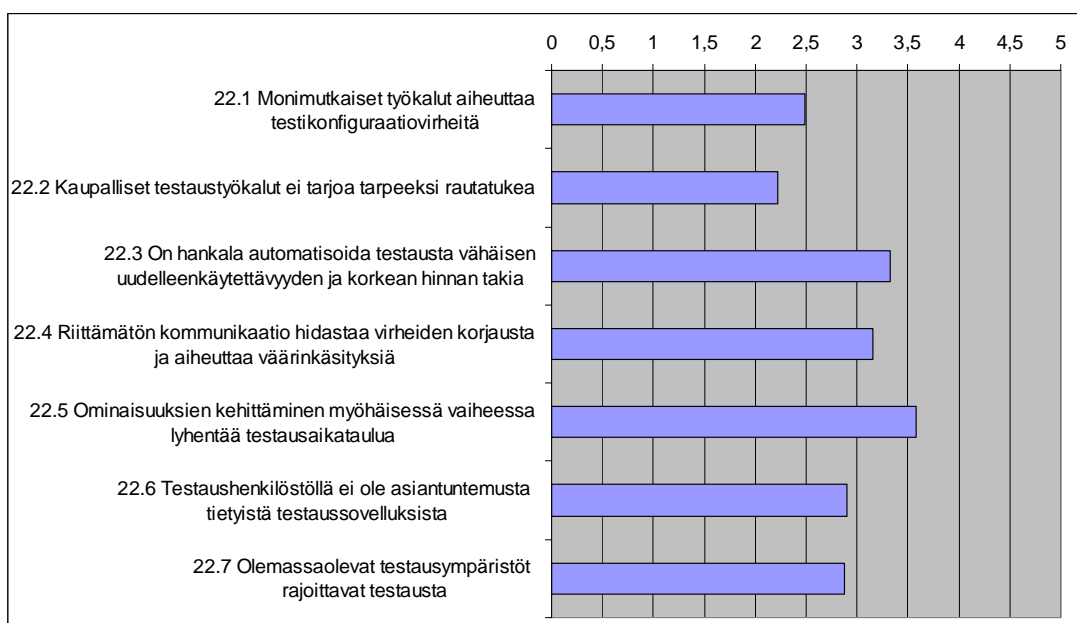


**Kuva 42. Laatuattributteihin liittyvien osakysymysten vastausten jakaumat**

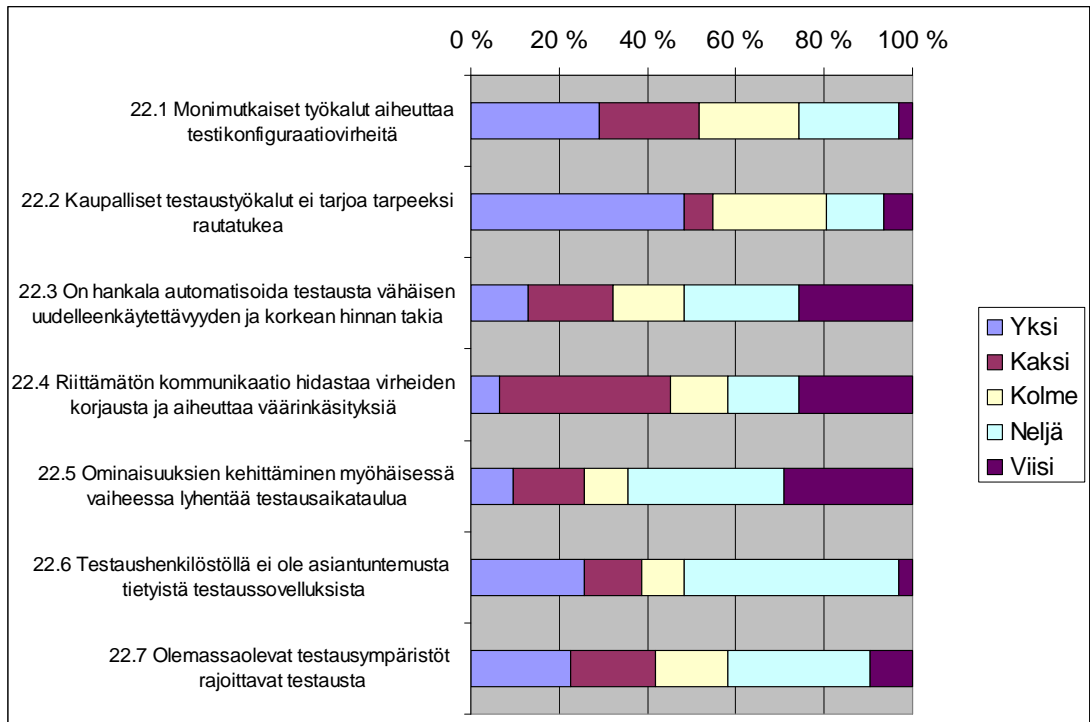
## 10.5 Ongelmat ja parannusehdotukset

Haasteltavia pyydettiin arvioimaan testauksen ongelmia 5-portaisella Likert-asteikolla vastaamalla testauksen ongelmiin liittyviin väittämiin. Vastaus 1 merkitsi täysin eri mieltä ja 5 täysin samaa mieltä. Kuvassa 43 on esitetty osakysymysten keskiarvo ja kuvassa 44 osakysymysten jakauma.

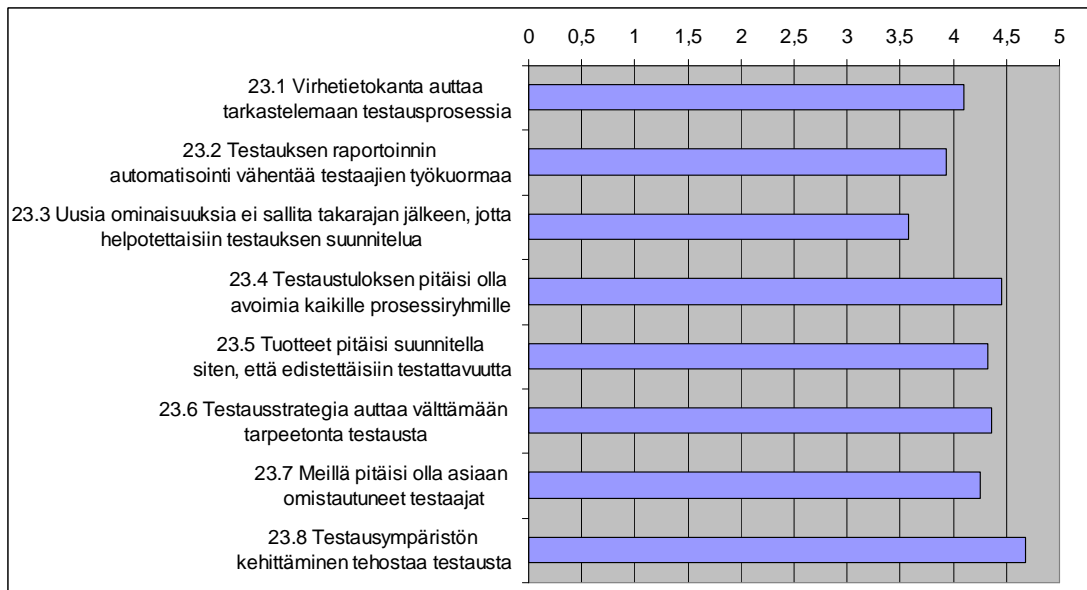
Haastatelluissa organisaatioyksiköissä suurimmiksi ongelmiksi arvioitiin testauksen automatisoinnin hankaluus vähäisen uudelleenkäytettävyyden (keskiarvo 3.3) ja korkean hinnan takia sekä ominaisuuksien kehittäminen myöhäisessä vaiheessa lyhentää testausaikataulua (keskiarvo 3.6). Pienimmiksi ongelmiksi haastateltavat arvioivat monimutkaisten testityökalujen aiheuttamat testikonfiguraatiovirheet (keskiarvo 2.5) sekä kaupallisten testityökalujen puutteellisen laitteen (keskiarvo 2.2).



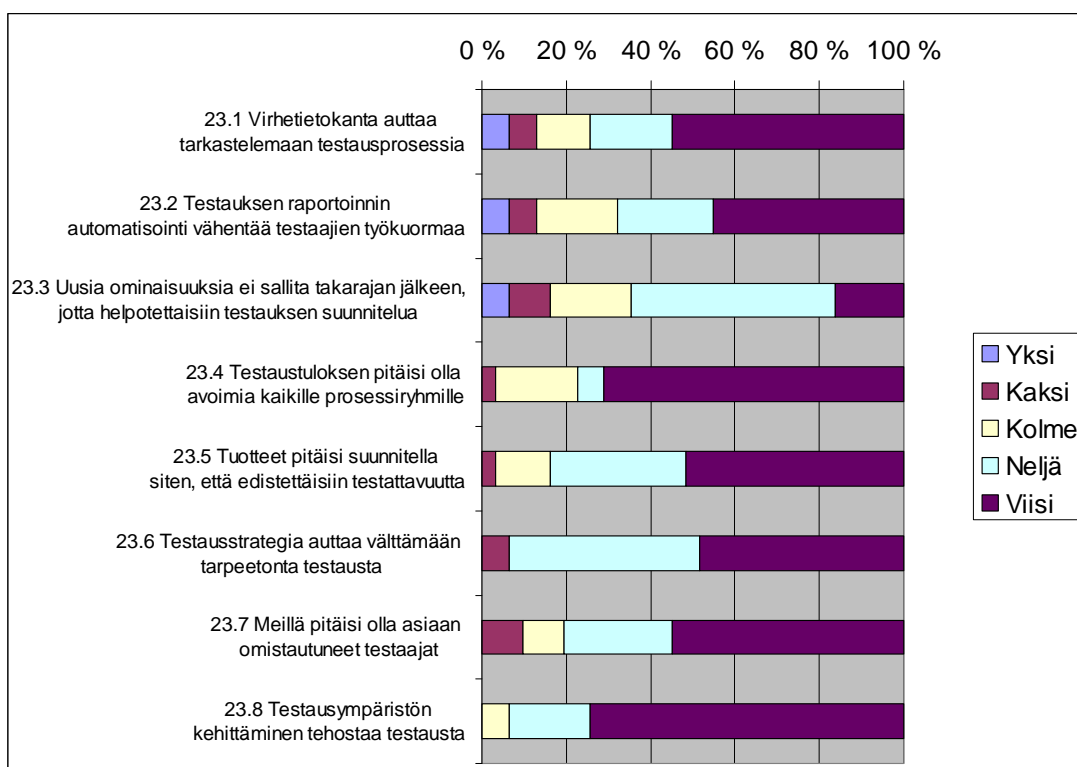
Kuva 43. Ongelmiin liittyvien osakysymysten vastausten keskiarvot



**Kuva 44. Ongelmiin liittyvien osakysymysten vastausten jakaumat**



**Kuva 45. Parannusehdotuksiin liittyvien osakysymysten vastausten keskiarvot**



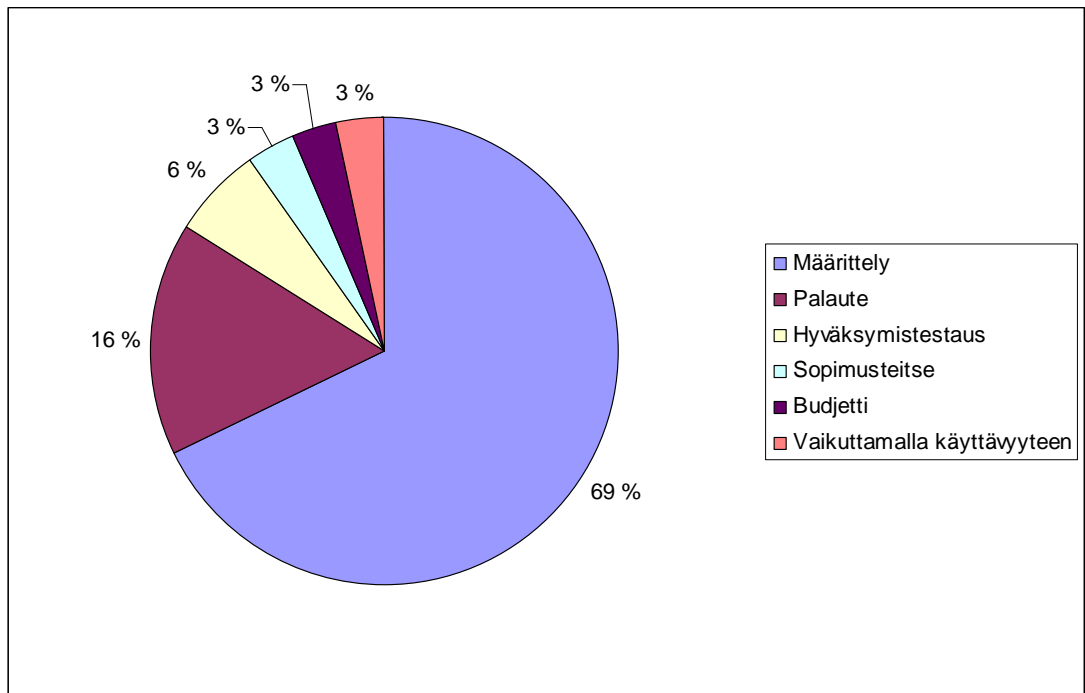
**Kuva 46. Parannusehdotuksiin liittyvien osakysymysten vastausten jakaumat**

Haastatelluissa organisaatioyksiköissä parannusehdotusten keskiarvot olivat pääpiirteissään neljän molemmin puolin (kuvat 45 ja 46). Kaikkein merkittävimäksi parannusehdotukseksi kuitenkin arvioitiin testausympäristön kehittäminen, joka auttaa testausta (keskiarvo 4.7) ja vähiten merkittäväksi toteutettavien ominaisuuksien kiinnittäminen tiettyyn ajanhetkeen (keskiarvo 3.6), jotka erottuivat selkeästi muista parannusehdotuksista.

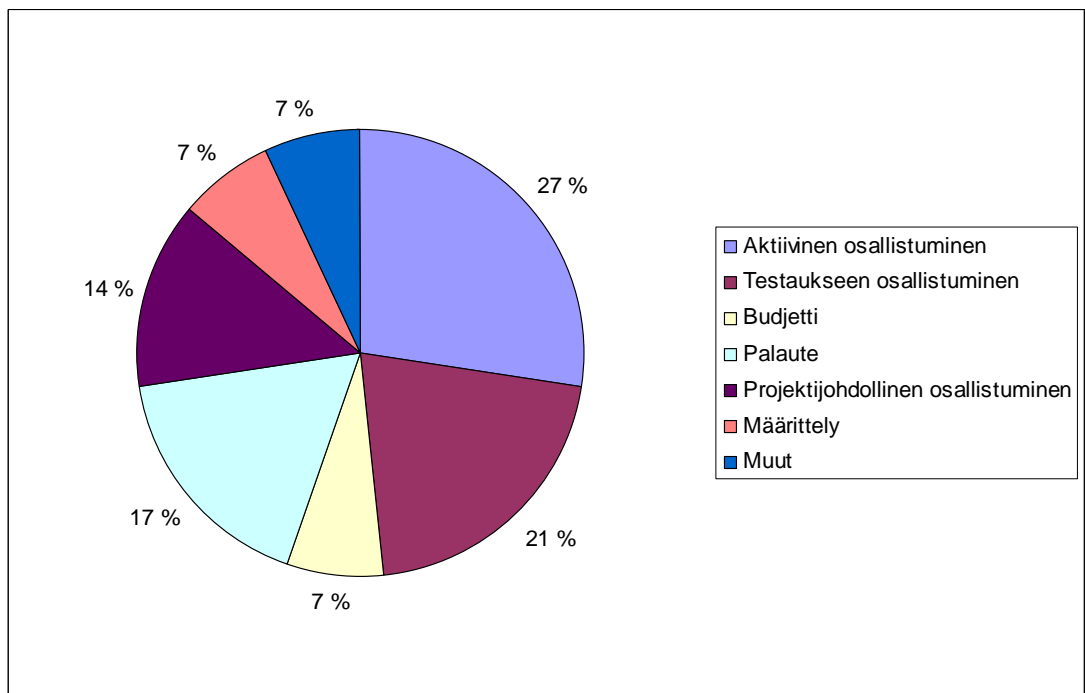
### **10.6 Merkittävimmät tavat asiakkaan ohjelmiston laatuun vaikuttamiseksi**

Kyselyssä vastaajia pyydettiin arvioimaan kolme merkittävintä tapaa tärkeysjärjestyksessä, miten asiakas voi vaikuttaa ohjelmiston laatuun. Kuvassa 47 on esitetty vastausten jakauma. Ylivoimaisesti merkittävimäksi tavaksi arvioitiin määrittely (69 %). Muita merkittävimmiksi arvioituja tapoja olivat palaute (16 %), hyväksymistestaus (6 %), sopimus (3 %), budjetti (3 %) ja vaikuttamalla käytettävyyteen (3 %).





**Kuva 47. Asiakkaan tehokkain keino ohjelmiston laatuun vaikuttamiseksi**

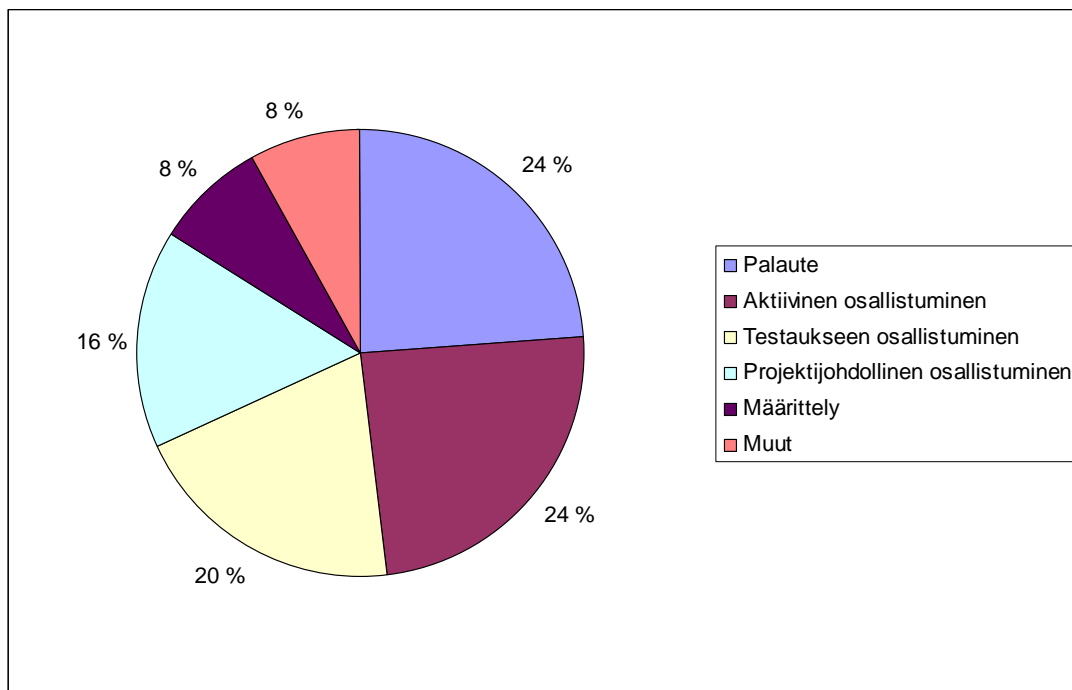


**Kuva 48. Asiakkaan toiseksi tehokkain keino ohjelmiston laatuun vaikuttamiseksi**

Toiseksi tehokkaimman tavan arvioinnissa oli jo huomattavasti enemmän hajontaa, kuva 48. Näistä merkittävimäksi tavaksi arvioitiin asiakkaan aktiivinen osallistu-

minen (27 %). Muita tapoja olivat asiakkaan testaukseen osallistuminen (21 %), asiakkaan antama palaute (17 %), projektijohdollinen työskentely (14 %) ja määrittely (7 %). Kategorian ”muut” (7 %) kuuluvat kaksi yksittäistä vastausta jotka ovat selkeiden pelisääntöjen määrittely ja negatiivisena puolena tuotteen ominaisuuksien vääristyminen yhden asiakkaan tarpeiden johdosta.

Kolmanneksi tehokkaimmaksi tavaksi arvioitiin seuraavia asioita (kuva 49): asiakaspalaute (24 %), asiakkaan aktiivinen osallistuminen (24 %), testaukseen osallistuminen (20 %), projektijohdollinen osallistuminen (16 %) ja määrittely (8 %). Kategoriiaan muut (8 %) kuuluu kaksi vastausta, jotka ovat asiakkaan laadunvarmistussuunnitelmat sekä käyttäjästatistiikka, joka muodostuu asiakkaan käyttäessä järjestelmää.



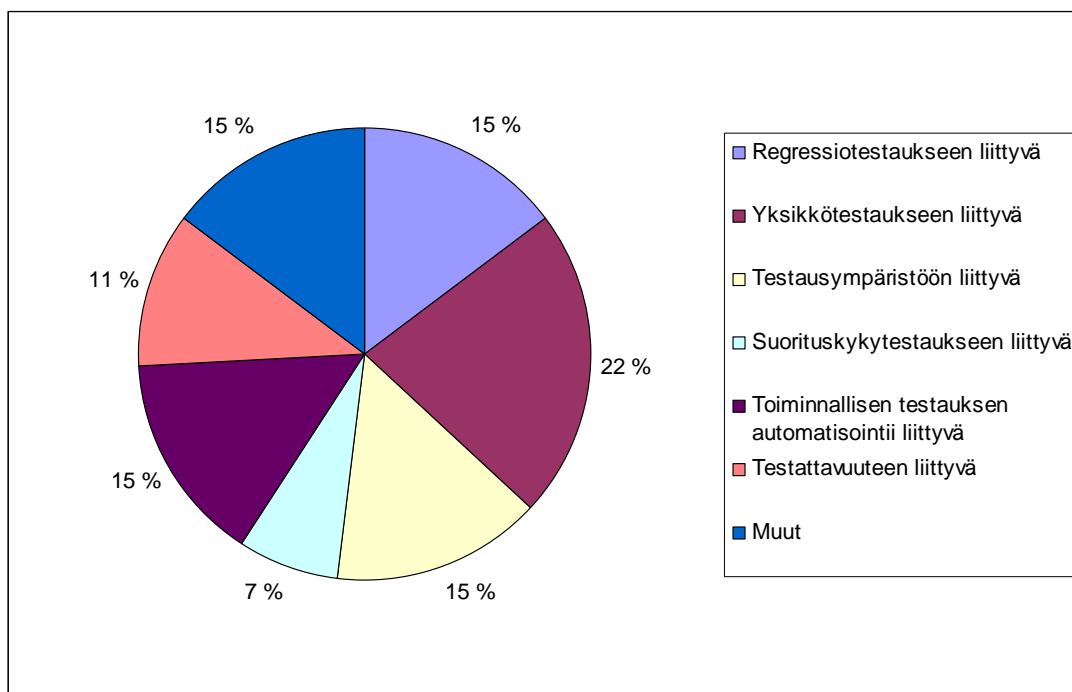
Kuva 49. Asiakkaan kolmanneksi tehokkain keino ohjelmiston laatuun vaikuttamiseksi

### **10.7 Tehokkaimmat testausautomaatioon liittyvää työkalua tai menetelmää**

Kyselyssä pyydettiin vastaajia arvioimaan kolme tehokkainta työkalua tai menetelmää liittyen testausautomaatioon tärkeysjärjestyksessä. Luokittelu on tehty Kitin kir-

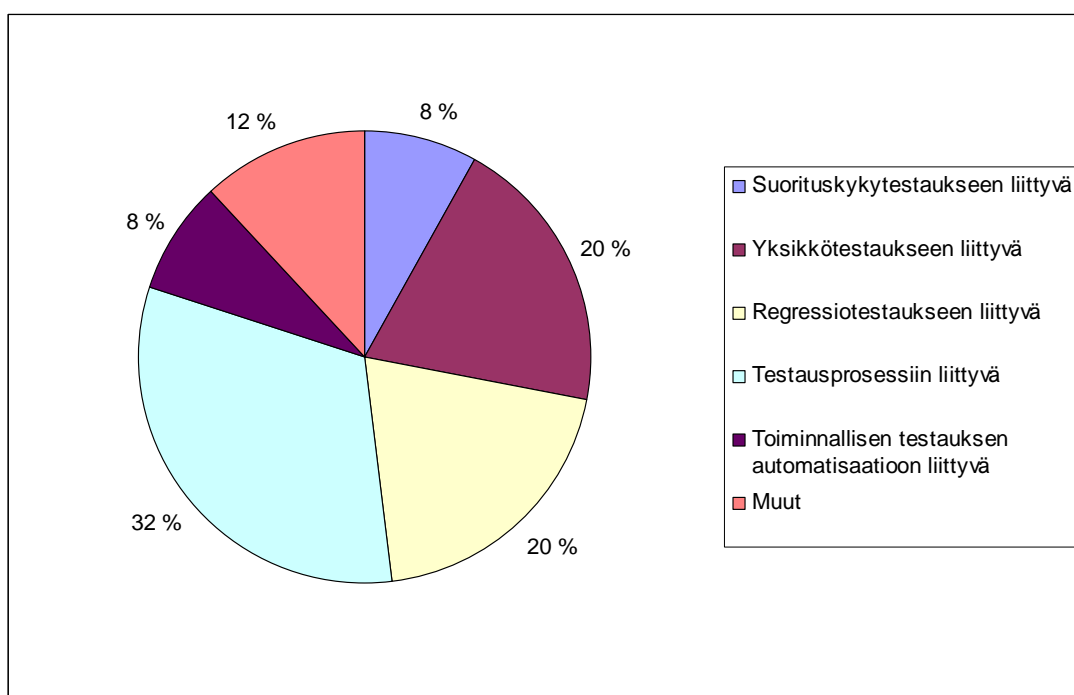
jan perusteella soveltaen muodostettuihin luokkiin, (Kit, 1995). Huomionarvoista on, että haastateltavista yrityksistä vain 15 (15/31 = 48 %) pystyi täyttämään kaikki 3 kohtaa. Muutamalla henkilöllä tämä johtui siitä, että heillä ei ollut lainkaan testauskokemusta, jolloin kaikki 3 kohtaa jäivät tyhjäksi. Muilla vastaukset tyhjäksi jättäneillä yrityksillä tämä voisi merkitä sitä, että testausautomaation tehokkuutta parantavia mahdollisuuksia ei tunneta kovin hyvin.

Tulokset hajaantuivat melko tasaisesti eri luokkien välille, mutta eniten mainintoja (kuva 50) saivat yksikkötestaukseen liittyvät työkalut tai menetelmät (22 %). Tähän luokkaan kuuluu muun muassa jatkuva integrointi ja TDD. Toiseksi eniten vastauksia tuli regressiotestauksen liittyviin, toiminnallisen testauksen automatisointiin liittyviin sekä testausympäristöön liittyviin työkaluihin ja menetelmiin, kuhunkin (15 %). Muita mainittuja työkaluja ja menetelmiä olivat testattavuuteen liittyvät (11 %) sekä kategoria ”muut” (15 %), joka koostuu neljästä vastauksesta. Nämä olivat standardinmukaisuustestaustyökalut, TTCN-3, kokonaisvaltaiset testauksen hallinnan työkalut sekä testauksen arvostuksen lisääminen yksikkö- ja integraatiotestauksessa.



**Kuva 50. Tehokkain menetelmä tai työkalu liittyen testausautomaatioon**

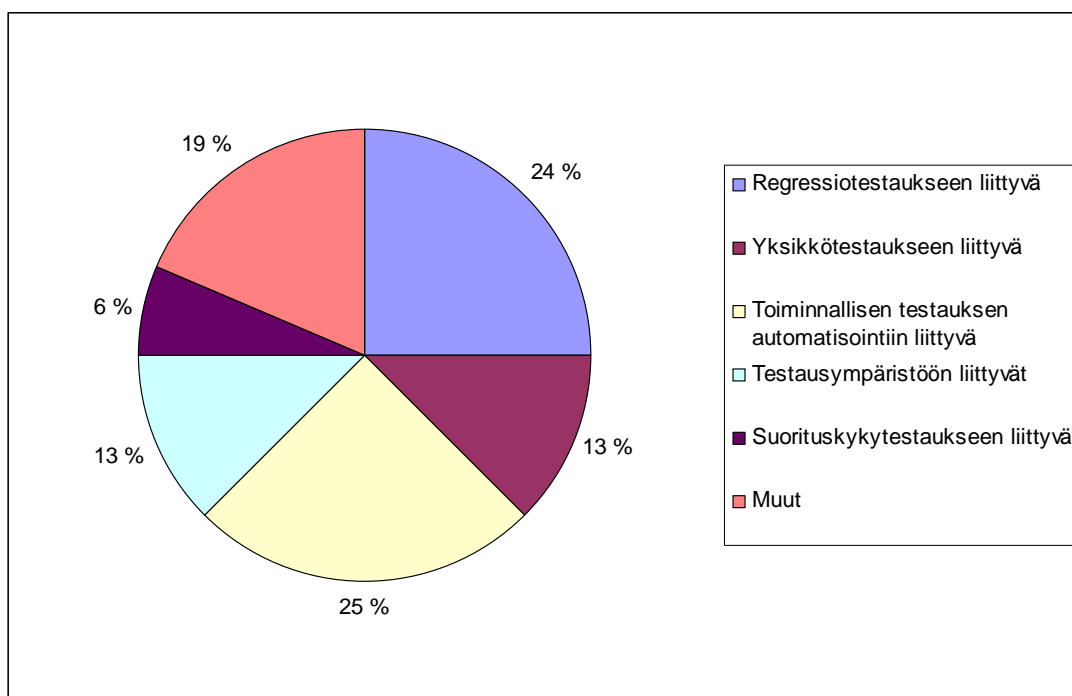
Toiseksi tehokkaimmaksi tavaksi (kuva 51) eniten kannatusta saivat testausprosessiin liittyvät työkalut tai menetelmät (32 %). Tähän luokkaan kuuluivat esimerkiksi hyvät työkalut ja testauksen fokusointi. Toiseksi eniten mainintoja keräsi yksikkötestaukseen ja regressiotestaukseen liittyvät työkalut ja menetelmät, molemmat (22 %). Muita identifioituja luokkia olivat toiminnallisen testauksen automaatioon liittyvät työkalut ja menetelmät (8 %) sekä luokka ”muut” (12 %). Viimeksi mainittuun luokkaan kuuluivat kolme yksittäistä vastausta, jotka ovat testauksen arvostuksen kohottaminen, modulaarinen mahdollisuus ottaa testausautomaatio käyttöön sekä testausautomaatio, joka minimoi manuaalisen työn määrän.



**Kuva 51. Toiseksi tehokkain menetelmä tai työkalu liittyen testausautomaatioon**

Kolmanneksi tehokkaimmaksi tavaksi (kuva 52) useimpia mainintoja keräsi toiminnallisen testauksen automatisointiin liittyvät työkalut tai menetelmät (25 %). Lähes saman prosenttiosuuden (24 %) keräsi regressiotestaukseen liittyvät työkalut tai menetelmät. Muita identifioituja kategorioita olivat testausympäristöön ja yksikkötestaukseen liittyvät työkalut tai menetelmät, molemmat (13 %), suorituskykytestaukseen liittyvät työkalut tai menetelmät (6 %) sekä kategoria muut (19 %). Tämä ryhmä koostui kolmesta yksittäisestä vastauksesta, jotka olivat automaation tuki osaamis-

mielessä, koodikatselmointi sekä negatiivisena puolena resurssit automaatiotyökalun hankkimiseen.



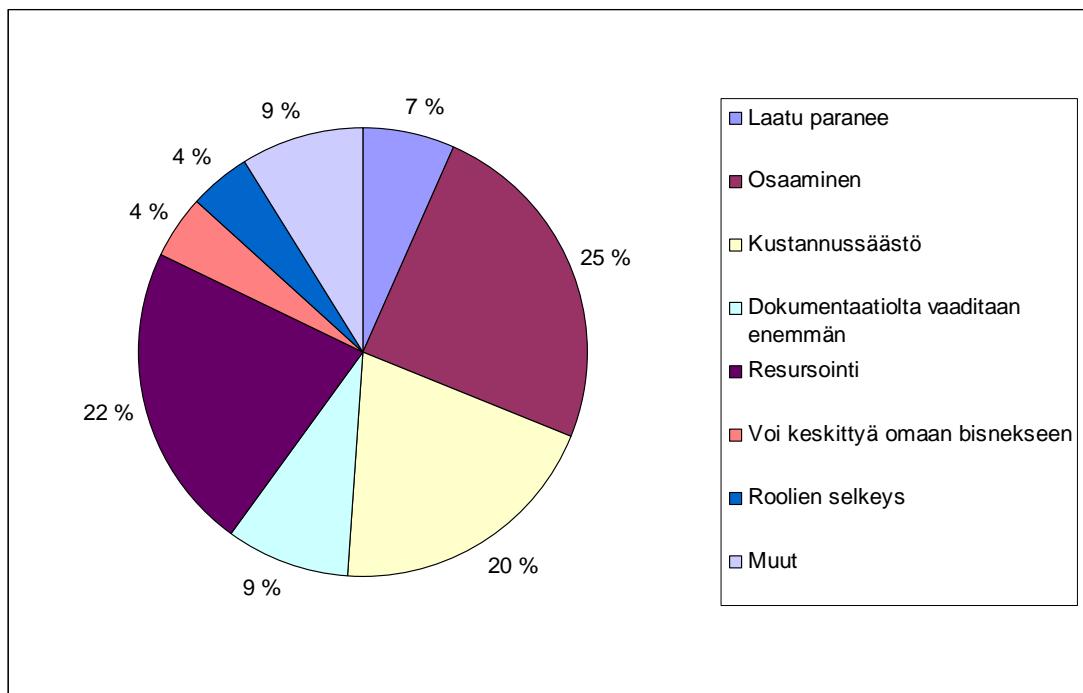
**Kuva 52. Kolmanneksi tehokkain menetelmä tai työkalu liittyen testausautomaatioon**

Yhteenvedona voi sanoa, että haastattelussa ei tullut esille mitään ylivoimaisen tehokasta testausautomaatiotyökalua tai menetelmää vaan vastaukset jakautuivat suhteellisen tasaisesti 8 identifioidun kategorian välille.

### **10.8 Tärkeimmät hyödyt ja haitat ohjelmistokehityksen ja –testauksen ulkoistamisessa**

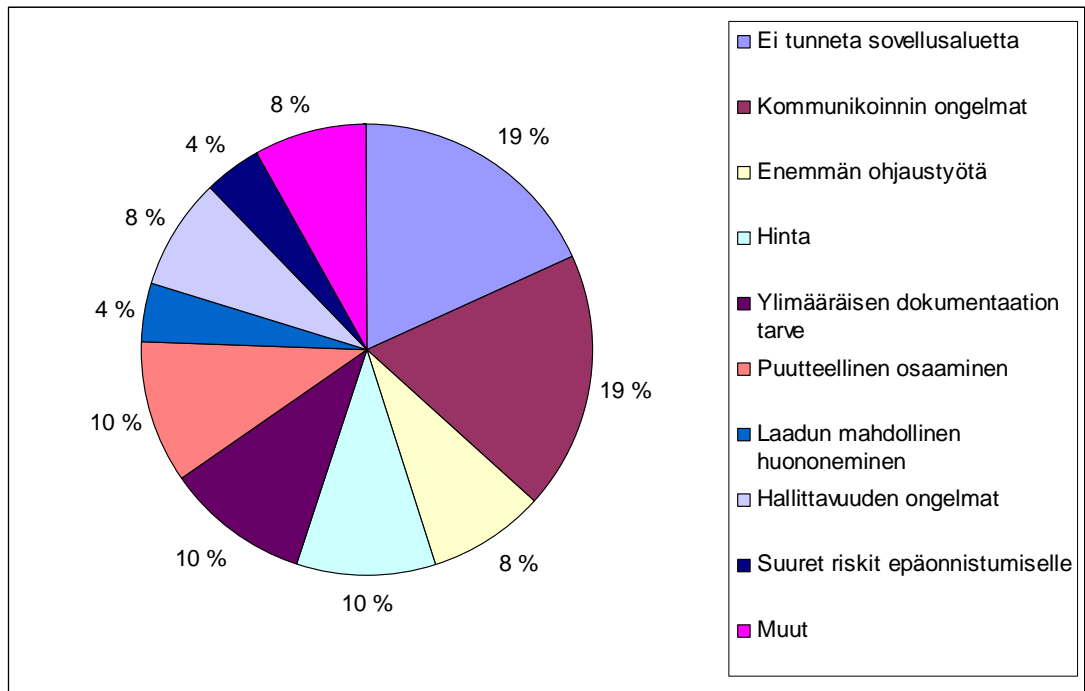
Viimeisenä kysymyksenä kyselyssä pyydettiin vastaajia nimeämään ja selittämään kolme kaikkein tärkeintä hyötyä ja haittaa ohjelmistokehityksen ja –testauksen ulkoistamisessa. Luokittelu on tehty soveltaen Ozan väitöskirjan perusteella muodostettuihin luokkiin, (Oza, 2006). Suurimmaksi hyödyksi (kuva 53) haastateltavat näkivät ulkopuolisen osaamisen hankkimisen (25 %). Muita usein mainittuja hyötyjä olivat: resursoinnin joustavuus (22 %) ja kustannussäästö (20 %). Näiden lisäksi mainittiin laadun parantuminen (7 %), keskittyminen omaan ydinliiketoimintaan (4 %), roo-

lien selkeys (4 %) ja, että dokumentaatiolta vaaditaan enemmän (9 %). Viimeksi mainitulla tarkoitetaan, että esimerkiksi määrittelydokumentaatio ja testitapaukset tulee tehtyä tarkemmin joka näkyy parempana lopputuloksena. Kategoriaan muut (4 %) kuuluu neljä hyötynäkökulmaa, jotka ovat: joustavuus teknologiassa, kiinteät kulut, parempi palvelu ja uusien työtapojen oppiminen.



**Kuva 53. Ohjelmistokehityksen ja -testauksen ulkoistamisen hyödyt**

Haittapuolista suurimmaksi haitaksi (kuva 54) arvioitiin osaamisen puute sovellusalueella sekä kommunikaation ongelmat, molemmat (19 %). Muita mainittuja haittoja ovat: kallis hinta (10 %), ylimääräisen dokumentaation tarve (10 %), puutteellinen osaaminen (10 %), enemmän ohjaustyötä (8 %), hallittavuuden ongelmat (8 %), laadun mahdollinen huononeminen (4 %) sekä suuret riskit epäonnistumiselle (4 %). Kategoriaan muut (8 %) kuuluu neljä yksittäistä vastausta, jotka ovat: ristiriitaiset intressit, kustannusten seurannan hankaluus, ongelmat luottamuksen kanssa sekä ongelmat aikataulutuksen kanssa projektin eläessä.



**Kuva 54. Ohjelmistokehityksen ja -testauksen ulkoistamisen haitat**

Huomionarvioista tuloksissa on se, että samat asiat on mainittuna sekä hyötynä, että haittana (hintaa, osaaminen, laadun parantuminen/huonontuminen, lisädokumentointi-tarve).

## 11 TULOSTEN POHDINTAA

Tutkimuksen perusteella korrelaatiota laadun kanssa ei havaittu liiketoimintasuuntautuneisuuden, ohjelmistokehitysmenetelmän ja kriittisyyden välillä.

Liiketoimintasuuntautuneisuuden ja ohjelmistokehitysmenetelmän osalta ei teoria antanut perustetta olettaa, että korrelaatiota löytyisi. Ainoastaan tiedettiin, että niillä saattaa olla jotain vaikutusta laatuun koska molemmat vaikuttavat ohjelmistokehitysprosessiin – ohjelmistokehitysmenetelmä välittömästi ja liiketoimintasuuntautuneisuus välillisesti. Tämän tutkimuksen mukaan niillä ei kuitenkaan ole vaikutusta laatuun.

Kriittisyyden osalta teorian perusteella oletettiin, että korrelaatiota laadun kanssa löytyisi. Korrelaation puuttuminen johtuu luultavasti tutkimuksen populaation määrittämisestä, joka määritteli populaation koostuvan keskikriittisiä sovelluksia tekevistä organisaatioyksiköistä. Näin ollen tutkimusdatasta puuttuu ääripäät eli todella kriittisiä ja yksikertaisia sovelluksia tekevät organisaatioyksiköt. On hyvä huomata, että tutkimuksessa havaittiin pientä korrelaatiota kriittisyyden ja laadun välillä, mutta se ei ole tilastollisesti 0.05 tai sitä tiukemmalla riskitasolla merkittävää. Mikäli ääripäät olisivat olleet tutkimusdatassa mukana, olisi korrelaatiota luultavasti havaittu.

Muilla tutkimushypoteeseilla (testauksen taso, standardin toteutuminen, asiakasosallistuminen, luottamus ja asiakassuhde) korrelaatiota laadun kanssa havaittiin teorian mukaisesti.

Yhtenä ongelmana tutkimuksessa oli se, että kaikkien vastaajien vastuualue yrityksessä ei ollut sama. Henkilöillä, jotka vastaavat ohjelmistokehitysprosessista saattaa olla luonnollinen intressi arvioida tuotettavan ohjelman laatu hieman todellisuutta paremmaksi, koska heidän päätöksiinsä yrityksen kehitysprosessi viimekädessä perustuu. Vastaavasti testauksesta vastaava henkilö saattaa vastata kysymyksiin hieman alakanttiin, mikäli hän ei itse vastaa kehitysprosessin kehittämisestä. Tämä johtuu



siitä että toisen tekemästä asiasta, tässä tapauksesta prosessista, on helpompi löytää kritisoitavaa kuin omasta tuotoksesta.

Myös kysymyslomakkeen tarkempi testaus olisi mahdollisesti parantanut kysymystenasettelua. Nyt kysymykset jäivät hieman monitulkintaisiksi. Esimerkiksi parannusehdotuksiin ja ongelmiin liittyvissä kysymyksissä jäi epäselväksi, pitääkö vastata henkilökohtaisena mielipiteenä vai miten firmassa asiat hoidetaan. Jotkut haastateltavat organisaatioyksiköt vastasivat kysymykseen oikopäätä eikä haastattelija aina huomannut varmistaa, miten vastaaja kysymyksen oli ymmärtänyt.

Joidenkin kysymysten termien laajuus ei myöskään ollut selvä. Esimerkiksi mitä vaiheita sana ”ohjelmistokehitys” sisältää. Kysymyksissä ei ollut määritelty sisältääkö termi myös esimerkiksi prosessiin sulautetut laadunvarmistusjärjestelmät. Kyselyn pituuden vuoksi kysymyksiä ei ollut aikaa määrittää riittävän täsmällisesti mikä tietysti aiheuttaa riskin monitulkintaisuudesta. Toisaalta liika määrittely tekee kyselystä pitkäväteisen, eikä se siltikään takaa että haastateltava henkilö ymmärtää termin merkityksen oikein.

Kysymyslomake oli laadittu englanniksi, mutta kysymykset esitettiin suomeksi. Se voi aiheuttaa ongelman, että kysymyksiä ei välttämättä kysytä joka kerta samalla tavalla. Kun haastatteluja tehtiin yli kolmekymmentä, voi ensimmäisen ja viimeisen haastattelun kysymykset olla kysytyt eri tavoilla. Nauhalta kysymysten esitystä kuunneltaessa ei tätä ongelmaa kuitenkaan havaittu siten, että se olisi vaikuttanut vastauksiin.

Myös ”asiakas” käsitteen tulkinta todettiin hankalaksi. Esimerkiksi useilla yrityksellä (erityisesti jotka tekevät tuotetta) on oma sisäinen asiakkaansa, joka ”edustaa” kehitettävän ohjelman käyttäjiä. Pitäisikö asiakas siis olla ulkoinen asiakas vai sisäinen asiakas? Haastatelluissa tein tulkinnan, että kyseessä on nimenomaan sisäinen asiakas. Tässä tulkinnassa on kuitenkin se vaara, että kehittäjiä on hyvin hankala tietää, kuinka hyvin sisäinen asiakas edustaa ulkoista asiakasta.

Kysymyslomakkeen laajuudesta johtuen kyselyyn vaadittaisiin periaatteessa kolme osallistujaa: henkilö, joka tuntee ohjelmistokehitysprosessin hyvin (projektipäällikkö), henkilö, joka tuntee testauksen käytännöt hyvin (testauspäällikkö) sekä sisäinen asiakas niiden yritysten tapauksessa, joissa sisäinen asiakas on käytössä. Sisäinen asiakas vaadittaisiin siksi, että hän pystyisi selvittämään kuinka hyvin kommunikointi asiakkaan kanssa oikein toimii. Hän kuitenkin edustaa ulkoisia asiakkaita ja mikäli kommunikaatio toimii huonosti, niin myös hänen tekemänsä päätökset asiakkaan puolesta saattavat olla huonoja.

Myös ketteriä menetelmiä soveltaville osa kysymyksistä ei olleet kovin relevantteja. Esimerkkinä mainittakoon kysymykset, jotka liittyivät asiakkaan osallistuneen eri ohjelmistokehitysprosessin vaiheissa. Vaiheet olivat jaettu määrittely/suunnittelu/toteutus/testaus eli vesiputousmallin mukaisesti. Ketteriä menetelmiä soveltavissa yrityksissä tuollaisia vaiheita ei ole joten, siitä seurasi että he joutuivat vastaavat kysymyksiin soveltaen kukin tavallaan. Mielestäni ketteriä menetelmille soveltaville yrityksille olisi nämä kysymykset pitänyt olla omansa tai ainakin nykyisille kysymyksille tulkintaohjeet ketteriä menetelmiä soveltavien yritysten tapauksessa.

Lisäksi kysymyslomakkeen kysymykset eivät soveltuneet kaikille haastatelluille yrityksille. Kysymyslomakkeen kysymykset olettivat, että haastatteluun osallistuva yritys vastaa kaikista ohjelmistokehitysprosessin osista itse ja että haastateltava henkilö tuntee kaikki osa-alueet. Kyselyyn osallistui muutama testausyritys, joiden liiketoiminta on tehdä ainoastaan yhtä palaa ohjelmistokehitysprosessista (testausta). Lisäksi muutama testauksesta vastaava henkilö ei tuntenut kokonaisprosessia kunnolla, joten muuhun kuin testaukseen liittyviin kysymyksiin vastaus perustui mutua-tuntumaan.

## 12 YHTEENVETO

Tutkimuksen perusteella laatu korreloi tilastollisesti merkittävästi testauksen tason, standardin toteutumisen, asiakasosallistumisen suunnitteluvaiheessa, asiakasosallistumisen ohjaukseen, luottamuksen sekä asiakassuhteeseen liittyvistä väittämistä väittämän "Väite 16.5: Asiakkaamme kommunikoi ja tekee yhteistyötä kiitettävällä tasolla meidän kanssa". Korrelaatio on positiivista, kuten hypoteesien teoreettinen perustelu oletti.

Regressioanalyysin perusteella regressiomalliksi saatiin laadun riippumisen standardin toteutumisesta, asiakasosallistumisesta suunnitteluvaiheessa sekä luottamuksesta. Mallin selitysasteeksi saatiin 54,0 %, jota voi pitää varsin hyvänä. Kuitenkin 46% laadun varianssista jää mallissa selittämättä. Mallin selittävästä varianssista 27,1 % selittää standardin toteutuminen, 16,3 % asiakasosallistuminen suunnitteluvaiheessa sekä 19,4 % luottamus.

Mallin mukaan laatua voidaan parantaa standardin noudattamisella, tiivistämällä asiakasyhteistyötä suunnitteluvaiheessa ja parantamalla luottamusta asiakkaan ja toimittajan välillä. Malli soveltuu tutkimuksen rajauksen mukaisissa organisaatioyksiköissä käytettäväksi.

Standardin toteutumista voidaan parantaa testauspolitiikalla, testausstrategialla, testauksen hallinnalla ja testauksen paremmalla suorittamisella. Asiakasosallistumista suunnitteluvaiheessa voidaan parantaa ottamalla asiakas mukaan käyttöliittymäsuunnitteluun, kehittämällä asiakkaalle prototyyppi, antamalla asiakkaan määrittää järjestelmän kontrollit ja turvakäytännöt sekä antamalla asiakkaan määrittää ja katselmoida teknisiä suunnitelmia. Mallin luottamus-muuttujaa voidaan parantaa edistämällä molemminpuolista luottamusta.

Tutkimuksen mukaan asiakas voi vaikuttaa laatuun eniten määrittelyn sekä palautteen kautta. Muiksi merkittäväksi tavoiksi haastateltavat arvioivat aktiivisen osallis-

tuminen projektityöhön, testaukseen osallistumisen sekä projektijohdollinen osallistuminen.

Tehokkaimmiksi testausautomaation liittyviksi työkaluiksi tai menetelmiksi haastateltavat arvioivat yksikkö- ja regressiotestaukseen sekä testausympäristöön liittyvät työkalut tai menetelmät. Muiksi merkittäviksi tavoiksi haastateltavat arvioivat suorituskykytestaukseen, toiminnalliseen testaukseen sekä testausprosessiin liittyvät työkalut tai menetelmät.

Ohjelmistokehityksen tai testauksen ulkoistamisessa merkittävimmäksi hyödyksi arvioitiin osaamisen hankkiminen, kustannussäästöt ja resursoinnin helpottuminen. Suuremmiksi haitoiksi haastateltavat arvioivat sovellusalueen tuntemisen puutteen ja kommunikoinnin ongelmat.

## LÄHTEET

- BACH, J. (2003) Exploratory Testing Explained, saatavissa: <http://www.satisfice.com/articles/et-article.pdf>, viitattu: 01.02.2009
- BASKERVILLE, R. L. & LEE, A. S. (2003) Generalizing Generalizability in Information Systems Research. *Information Systems Research*, 14, 23.
- BENDER, D. (1993) *Writing Testable Requirements*, Software Testing Analysis & Review (STAR) Conference Proceedings.
- BENTON, W. C. & MALONI, M. (2004) The influence of power driven buyer/seller relationships on supply chain satisfaction. *Journal of Operations Management*, 22.
- BOEHM, B. & TURNER, R. (2004) Balancing agility and discipline: A guide for the perplexed. Addison-Wesley.
- DYBÅ, T. (2000) An Instrument for Measuring the Key Factors of Success in Software Process Improvement. *Empirical Software Engineering*, 5, 357-390.
- FINK, A. (2003) *The Survey Handbook*, SAGE Publications, Inc.
- FINK, A. & KOSECOFF, J. (1985) *How to conduct surveys A Step-by-Step Guide*, Newbury Park, CA, SAGE Publications, Inc.
- FISHMAN, C. (2007) They Write the Right Stuff saatavissa: <http://origin-www.fastcompany.com/magazine/06/writestuff.html?page=0%2C0>, viitattu: 24.03.2009
- FRANK, L. (2004) *Tilastollisen analyysin perusteet: SPSS-moniste*.
- GUIMARAES, T., MCKEEN, J. D. & WETHERBE, J. C. (1994) The Relationship Between User Participation and User Satisfaction: An Investigation of Four Contingency Factors. *MIS Quarterly/December 1994*, 26.
- HAIKALA, I. & MÄRIJÄRVI, J. (2006) *Ohjelmistotuotanto*, Helsinki, Talentum.

HENDRICKSON, E. (2005) Agile Testing, saatavissa:

<http://video.google.com/videoplay?docid=-3054974855576235846&ei=scmKScyIMZ-i2wL7sfG6Cw&q=agile+testing&hl=fi>, viitattu: 10.02.2009

HENDRICKSON, E. (2008) Agile Testing Overview, saatavissa:

[www.testobsessed.com/wordpress/wp-content/uploads/2008/08/AgileTestingOverview.pdf](http://www.testobsessed.com/wordpress/wp-content/uploads/2008/08/AgileTestingOverview.pdf), viitattu: 20.02.2009

HIRSJÄRVI, S., REMES, P. & SAJAVAARA, P. (2004) *Tutki ja kirjoita*, Jyväskylä, Kustannusosakeyhtiö Tammi.

IEEE/ANSI (1983) *IEEE Standard for Software Test Documentation*.

ISO/IEC (2002) *ISO/IEC 15504-1, Information Technology - Process Assessment - Part 1: Concepts and Vocabulary*.

ISO/IEC (2008) CD 25010.2, Software engineering-Software product Quality Requirements and Evaluation (SQuaRE)Quality model. 42.

KETTUNEN, V. (2008) Laadunhallinta ketterissä menetelmissä. *Tietotekniikan osasto*. Lappeenranta, Lappeenrannan teknillinen yliopisto.

KIT, E. (1995) *Software Testing in the Real World: Improving the Process*, Reading, MA, Addison-Wesley.

KITCHENHAM, B. A., PFLEEGER, S. L., PICKARD, L. M., JONES, P. W., HOAGLIN, D. C., EMAM, K. E. & ROSENBERG, J. (2002) Preliminary Guidelines for Empirical Research in Software Engineering. *IEEE Transactions on Software Engineering*, 28, 721-733.

KOSKELA, L. (2007) Scrum: Ketterien menetelmien markkinajohtaja, saatavissa:

[http://ttry.fi-bin.directo.fi/@Bin/c3f89f97337d8e97152b4e3b4b9396ee/1233841389/appli-cati-](http://ttry.fi/bin.directo.fi/@Bin/c3f89f97337d8e97152b4e3b4b9396ee/1233841389/appli-cati-)

[on/pdf/11062393/04\\_ScrumMarketLeaderOfAgileMethods\\_handout\\_LasseKoskela.pdf](on/pdf/11062393/04_ScrumMarketLeaderOfAgileMethods_handout_LasseKoskela.pdf), viitattu: 05.02.2009

LITWIN, M. S. (1995) *How to Measure Survey Reliability and Validity*, Thousand Oaks, CA, Sage Publications.

MATTILA, M. (2008) KvantiMOTV - Regressioanalyysi, saatavissa: <http://www.fsd.uta.fi/menetelmaopetus/regressio/analyysi.html>, viitattu: 13.01.2009

NIKULA, U., PASI, A. & SMOLANDER, K. (2008) Customer-supplier Issues in Software Development. *European Conference on Information Systems (ECIS)*. Galway.

OLSON, M. H. & IVES, B. (1980) Measuring User Involvement in Information System Development. *Proceedings of the International Conference on Information Systems*. Philadelphia, PA.

OZA, N. (2006) An Empirical Evaluation of client-vendor Relationships in Indian Software Outsourcing Companies, saatavissa: [http://www.sbl.tkk.fi/nilayoza/finalversion\\_oct.pdf](http://www.sbl.tkk.fi/nilayoza/finalversion_oct.pdf), viitattu: 01.04.2009

PFLEEGER, S. L. & KITCHENHAM, B. A. (2001) Principles of Survey Research Part 1: Turning Lemons to Lemonade. *Software Engineering Notes*, 26, 16-18.

REID, S. (2008) The New International Testing Standard. Testing Solutions Group.

SAARANEN, A., PUUSNIEKKA, A., ESKOLA, J. & KUULA, A. (2008) Mittaaminen: Mittarin luotettavuus, saatavissa: <http://www.fsd.uta.fi/menetelmaopetus/mittaaminen/luotettavuus.html>, viitattu: 25.02.2009

SABHERWAL, R. (1999) The Role of Trust in Outsourced IS Development Projects. *Communication of the ACM*, 42, 8.

- SANTAMARIA, M. G. (2007) Agile & Scrum: What are these methodologies and how will they impact QA/testing roles? , 7.
- SIVONEN, J. (2004) KvantiMOTV - Korrelaatio ja riippuvuusluvut, saatavissa: <http://www.fsd.uta.fi/menetelmaopetus/korrelaatio/korrelaatio.html>, viitattu: 12.03.2009
- SOFTWARE ENGINEERING INSTITUTE, C. M. U. (2009) CMMi website, saatavissa: <http://www.sei.cmu.edu/cmami/>, viitattu: 28.03.2009
- SOMMERVILLE, I. (1995) *Software Engineering*, Essex, England, Addison Wesley.
- SPSS (2004) SPSS 16.0 for Windows. Chicago, SPSS Inc.
- TAIPALE, O. (2007) Observations on software testing practise. Lappeenranta, Lappeenranta University of Technology.
- TAIPALE, O., SMOLANDER, K. & KÄLVIÄINEN, H. (2006) A Survey on Software Testing. *6th International SPICE Conference on Software Process Improvement and Capability dEtermination (SPICE'2006)*. Luxembourg.
- WESSELIUS, J. & VERVERS, F. (1990) Some Elementary Questions on Software Quality Control. *Software Engineering Journal*, Vol.5.
- WIKSTRÖM, P. (2007) Agile Testing, saatavissa: <http://www.testway.se/download/Seminar9/Agile%20Testing.pdf>, viitattu: 06.02.2009